Prime-Only Summation and Nonredundant Sheaves: A Topos-Theoretic Basis for Measuring Independent Structure in Artificial Intelligence

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

February 9, 2026

Modern AI systems exhibit an explosion of observable behaviors, yet much of this apparent novelty may be recombination rather than the creation of genuinely new independent structure. This paper proposes a precise mathematical lens for separating irreducible degrees of freedom from composite redundancy by unifying two ideas: prime-only summation in number theory and generator-only aggregation in a Grothendieck topos of contextual behaviors. In arithmetic, primes index the irreducible generators of multiplicative composition, and prime-restricted sums isolate nonredundant contributions while treating composites as reconstructible configurations. We show how the same structural principle can be transported into an "AI topos" built from contexts, coverings, and sheaves of model responses, where nonredundant sheaves form an irredundant generating family for the entire behavioral universe. Within this framework, novelty becomes a generator-level functional rather than a count of distinct outputs, and configuration-level statistics become derived via Euler-product analogues whose logarithms expose generator-only content. The result is a falsifiable methodology: define a site of probes, infer a minimal generator basis, measure generator novelty and coupling, and test stability and reconstruction against strong null models. The paper closes with concrete experimental designs and prediction targets for evaluating whether independent structure in AI grows slowly, thins with scale, and is best tracked by generator-only sums.

Keywords: prime-only summation, Grothendieck topos, sheaf theory, irredundant generators, nonredundant sheaves, contextual behavior, AI evaluation, novelty functional, Euler product analogue, generator spectrum, model diagnostics, compositional structure, independence measurement, null models, reconstruction tests.

A collaboration with GPT-5.2-Thinking. 61 pages. CC4.0.

Outline

1. Introduction: From Prime Sums to Generator Sums in AI
1.1 The recombination problem in measuring AI novelty
1.2 Prime-only summation as a canonical nonredundant aggregator
1.3 The central thesis: independent reality equals generator content
1.4 What would make the thesis nontrivially true and testable
1.5 Roadmap of results, diagnostics, and experimental predictions

2. Mathematical Backbone: Irreducibles, Generators, and Nonredundancy
2.1 Primes as generators in a free commutative monoid
2.2 Nonredundant generating families in categories and topoi
2.3 Additivity and why "counting everything" overcounts
2.4 Canonical versus chosen bases: uniqueness and gauge freedom
2.5 The irreducible aggregation principle as a categorical template

3. The AI Topos: Contexts, Coverings, and Sheaves of Behavior
3.1 Contexts as probe-sets and refinement as information order
3.2 Coverings as "sufficient local evidence" and gluing conditions
3.3 The sheaf of responses: distributions, restrictions, and marginals
3.4 What "global behavior" means in a sheaf semantics
3.5 Practical construction from logs, benchmarks, and probes

4. Nonredundant Sheaves as Independent Degrees of Freedom
4.1 From representables to generators: how the basis arises
4.2 Irredundant generator selection and minimality criteria
4.3 Redundancy as reconstructibility and conditional dependence
4.4 Composite sheaves as glued configurations and emergent states
4.5 Stability of generator bases as a reality test

5. Prime-Only Summation Translated: Generator-Only Aggregation Functionals
5.1 Defining generator-only sums for categorical invariants
5.2 Additive and colimit-respecting functionals as admissible measures
5.3 Overcounting corrections: why composites must not be primitive
5.4 Multiple weightings: scale, impact, rarity, and intervention cost
5.5 Interpreting "prime-only" as "basis-only" without metaphor

6. Euler-Product Analogues: From Configuration Totals to Generator Spectra
6.1 Configuration spaces as multisets of generators
6.2 Generator energies and additive costs in the AI topos
6.3 Partition-like functionals and factorization under independence
6.4 Log-linearization: why logs expose generator-only structure
6.5 Interaction terms as controlled violations of factorization

7. Novelty and Surprise: Measuring Irreducible Growth in AI Systems
7.1 Generator activation and novelty events from observational updates
7.2 A topos-native novelty functional and finite-sample estimators
7.3 Distinguishing novelty from diversity in outputs and trajectories
7.4 Predicted regimes: early rapid discovery and long tail thinning
7.5 Coupling diagnostics: when novelty is interaction-driven

8. Model Selection and Diagnostics: Falsifying the Generator Ontology
8.1 Competing hypotheses: additive structure, random labeling, mixed regimes
8.2 Diagnostic plots: basis stability, spectra, and reconstruction error
8.3 Robustness to noise, missingness, and observational bias
8.4 Null models: permutation baselines and factor-scrambling analogues
8.5 Clear falsifiers: instability, nonreconstructibility, and null equivalence

9. Applications: Evaluation, Safety, and Systems Engineering Interpretations
9.1 Capability accounting: independent skills versus recombined outputs
9.2 Logging and telemetry: templates as generators and events as composites
9.3 Reliability and safety: independent failure modes and coupling growth
9.4 Semantics and cognition: irreducible causes versus derived narratives
9.5 Choosing generator energies and novelty weights in applied settings

10. Experimental Program: Turning Sheaf Bases into Measurable Predictions
10.1 Designing probe sites and coverages for real models
10.2 Estimating generator bases from interaction streams and benchmarks
10.3 Controlled interventions: changing temperature-like parameters
10.4 Prediction targets: scaling laws, stability, and reconstruction performance
10.5 Releasing benchmarks: reproducible protocols and reporting standards

11. Limitations and Open Problems: Beyond Unique Factorization
11.1 Where prime-factor intuition breaks in general topoi
11.2 Non-uniqueness of bases and invariants under basis changes
11.3 Beyond primes: irreducibles in other algebraic and categorical settings
11.4 Computational scaling: selection, estimation, and interaction learning
11.5 Theoretical frontier: cohomological obstructions as emergence measures

12. Conclusion: Prime-Only Summation as a Measure of Independent Reality in AI
12.1 Summary of claims and what is genuinely new
12.2 The irreducible aggregation lens as a unifying viewpoint
12.3 Final falsifiable signatures and the next research steps

References

Art

1. Introduction: From Prime Sums to Generator Sums in AI

The contemporary discourse around artificial intelligence is saturated with "novelty." New benchmarks appear weekly, models display surprising behaviors, and the observable surface of outputs seems to expand without limit. Yet there is a quiet structural question underneath the spectacle: how much of what we see is genuinely new independent capability, and how much is recombination of a smaller set of latent primitives? This paper argues that many disputes about "emergence," "capability," and "generalization" are, at bottom, disputes about *what should be counted as irreducible*. The proposed lens is intentionally austere: do not count everything that can be observed; count only what is nonredundant in the generative basis. In arithmetic, the prime numbers are the canonical example of irreducibles, and restricting summation to primes is the canonical move for isolating nonredundant generator contributions. The paper transports that move into an "AI topos" of contextual behaviors, where nonredundant sheaves form an irredundant generating family, and where generator-only aggregation becomes the correct primitive measurement of independent structure. The ambition is not metaphor. The ambition is a testable framework in which "independent reality" is operationally defined as generator content, and in which composite behaviors are treated as reconstructible consequences of gluing and composition rather than as new degrees of freedom.

1.1 The recombination problem in measuring AI novelty

A model's output space is combinatorial. Even a fixed vocabulary and a finite context window already permit an astronomical number of distinct strings. Add temperature variation, sampling noise, and prompt variation, and the apparent diversity becomes effectively unbounded. This creates an immediate danger for evaluation: naive novelty measures are almost guaranteed to report "newness" forever, even if the underlying mechanism is not expanding in any meaningful sense.

The same issue appears at multiple scales. At the level of text, "new sentences" are cheap. At the level of tasks, "new problem instances" are cheap. At the level of tool use and multi-step planning, "new trajectories" are cheap. In each case, the observable is a *configuration*: a composite object assembled from simpler parts—tokens, subskills, micro-decisions, latent features, or internal circuits. Counting configurations confuses *combinatorial multiplicity* with *independent capability*. A single primitive shift in internal representation can generate a thousand new surface behaviors, while an impressive surface diversity can arise from a relatively small set of reusable templates.

This recombination problem is not a minor accounting nuisance; it is the core reason that the field oscillates between "the model is just remixing" and "something genuinely new has emerged." Without a principled way to distinguish independent primitives from composite

manifestations, both intuitions can be correct at once, depending on what is being counted. The paper's starting point is therefore a demand: define novelty at the level of irreducible generators, not at the level of observable configurations.

1.2 Prime-only summation as a canonical nonredundant aggregator

Prime-only summation is a sharpened version of a simple principle: if composites can be generated from irreducibles, then summing over all integers overcounts the primitive structure many times. In the integers, primes are the irreducible generators under multiplication. Every positive integer decomposes uniquely into prime powers, so a function that is additive on prime exponents (or otherwise generator-additive) is naturally computed by aggregating over primes, not over all integers as if each were independent.

The exact arithmetic details are not the point here. The point is the structural role primes play: they represent the minimal basis elements under a composition rule with unique factorization. Once you accept that, a prime-restricted sum stops looking like a quirky restriction and starts looking like *the correct primitive accounting* for independent contributions. Composites are then understood as configurations whose statistics should be reconstructed from prime data through composition operators (Euler products being the canonical example). In other words, prime-only aggregation is not just "skipping numbers." It is a disciplined refusal to count derived structure as independent structure.

This paper uses prime-only summation as a conceptual prototype for what we want in AI evaluation: an aggregation rule that measures independent degrees of freedom once, and treats combinatorial recombination as derived. Primes are the mathematical training wheels for the more general idea: irreducible aggregation.

1.3 The central thesis: independent reality equals generator content

The central thesis can be stated in one sentence:

Independent reality, in an AI system, is the content of its nonredundant generators; everything else is composite and should be reconstructed rather than primitively counted.

To make that sentence meaningful, we must specify what "generator" means in AI, and we must avoid conflating it with an implementation detail. The proposal is to work in a topos of contextual behaviors: objects are sheaves encoding model behavior across contexts, morphisms are restriction/consistency maps, and covers express what counts as sufficient local determination of global behavior. In that setting, "nonredundant sheaves" are a generating family: a basis-like set of primitive behavioral objects from which all other behaviors can be built by gluing and colimits. They are "nonredundant" in the sense that removing any of them destroys the ability to reconstruct the behavioral universe.

Once such a generator family is identified, generator-only aggregation becomes the analogue of prime-only summation. It is the correct primitive measurement of independent structure for any functional that respects the topos's composition operations (coproducts, quotients, gluing colimits). Composite behaviors are not dismissed; they are relocated. They become emergent configurations whose abundance reflects combinatorial assembly, not necessarily the growth of new independent degrees of freedom.

This thesis has a useful moral edge: it is designed to prevent a common kind of epistemic inflation—mistaking the size of the behavioral surface for the dimensionality of the independent basis that supports it.

1.4 What would make the thesis nontrivially true and testable

A thesis like "independent reality equals generator content" can fail in two ways. It can fail conceptually by being too vague, or it can fail empirically by being false in real systems. The paper is built to avoid both failures by specifying a concrete set of conditions under which the generator-only lens is legitimate and by proposing diagnostics that can refute it.

Conceptual conditions include:

A defensible site of contexts and a defensible coverage. The topos construction requires a clear account of what contexts are (probe-sets, tasks, partial observations) and what coverings mean (which local views jointly determine a global behavior). If contexts and coverings are arbitrary, generator claims are arbitrary.

A meaningfully defined generator family. A generator family is not "whatever features you like." It must generate the category of behaviors under the allowed operations, and it must be irredundant or at least minimal in a specified sense. Different choices of generator families are allowed, but then the invariants must be stable under reasonable changes of basis or else the ontology is a coordinate artifact.

Admissible functionals. Prime-only aggregation is justified for quantities that are additive on generators and treat composites as derived. Similarly, generator-only aggregation is justified for functionals that respect the topos's composition operations. If your target quantity is dominated by interactions that do not reduce to generator content, then generator-only sums will be insufficient, and that insufficiency is itself diagnostic.

Empirical tests must include:

Stability tests. If generator-only measures reflect something real, they should be stable across observational windows and robust under reasonable reweightings and resamplings.

Reconstruction tests. One should be able to reconstruct composite statistics from generator-level spectra with controlled error, and the error should localize to identifiable interaction terms rather than being arbitrary.

Null-model separation. Generator-level structure must not persist under baselines that destroy meaningful compositional structure (label permutation, factor scrambling analogues, probe randomization), or else the signatures are artifacts.

In other words, the thesis becomes nontrivial when it yields measurable predictions that can fail, and when it provides a disciplined mechanism for telling "basis reality" apart from "configuration abundance."

1.5 Roadmap of results, diagnostics, and experimental predictions

The remainder of the paper unfolds in three arcs: formalization, measurement, and testing.

First, we formalize the irreducible aggregation principle across arithmetic and category theory. The number-theoretic side supplies a clean archetype: irreducibles, unique factorization, generator-additive quantities, and reconstruction of configurations from generator data. The categorical side generalizes this: in a Grothendieck topos, a generating family plays the role of irreducibles, and any colimit-respecting functional is determined by its values on generators. This establishes the sense in which "prime-only" becomes "generator-only."

Second, we build an explicit AI topos from contexts and coverages, define nonredundant sheaves as an irredundant generating family, and define generator-only aggregation functionals appropriate for AI evaluation. Two key constructions follow: a generator-spectrum viewpoint (an analogue of e(p) on primes) and a generator-novelty functional that measures irreducible novelty while correcting for configuration-level overcounting.

Third, we develop falsifiable diagnostics and an experimental program. Diagnostics include basis stability curves, reconstruction error as a function of basis size, response functions under controlled "temperature-like" parameters, and null-model baselines designed to break spurious structure. The central experimental predictions are qualitative but sharp: apparent novelty at the configuration level should grow rapidly even when generator novelty thins; generator spectra should stabilize early if the system is mostly recombinatory; and departures from factorization should appear as localized coupling terms rather than as diffuse failure if a generator ontology is approximately correct.

The payoff is a framework that forces clarity: either independent structure grows, and generator-only measures detect it robustly, or it does not, and the framework fails visibly. In both cases, the paper's contribution is the same: it relocates the conversation about AI novelty

from surface variety to irreducible accounting, and it supplies the mathematics needed to make that relocation testable.

2. Mathematical Backbone: Irreducibles, Generators, and Nonredundancy

The aim of this section is to make the paper's central translation precise: "prime-only summation" is not an arbitrary restriction, but the canonical way to aggregate independent structure when a system admits an irreducible generator basis and a composition rule that creates vast composite multiplicity. In arithmetic, this is embodied by primes and multiplication. In categorical settings, the same architecture appears as generating families and colimit-based construction of objects. The technical point is simple but far-reaching: once you separate a system into generators and configurations, the natural primitive sums live on generators, while configuration-level counts are derived quantities that often massively overcount the independence you actually care about.

2.1 Primes as generators in a free commutative monoid

Consider the positive integers under multiplication. This structure is a free commutative monoid on the set of primes. "Free" means there are no relations among generators beyond commutativity and associativity, and "on the primes" means every integer corresponds to a finite multiset of primes with multiplicity.

Concretely, each integer $n > 1$ has a unique representation as a product of primes with exponents. This unique factorization is the statement that the map from integers to prime-exponent vectors is injective and complete: an integer is exactly the data of how many times each prime appears. The integer thus becomes a configuration, and the primes become the independent modes or degrees of freedom.

This perspective is not merely philosophical. It tells you what kinds of functions and sums are structurally natural. Any quantity that is additive over prime exponents has a canonical generator-level expression. For example, if we assign each prime p a "cost" $e(p)$, then a composite integer has total cost equal to the sum of costs of its prime factors counted with multiplicity. The generator contributions are independent in the precise sense that the total cost is a linear functional on the occupancy vector. This is the archetype of the generator–configuration distinction.

The key lesson is that summing over all integers as if each were independent can be badly misleading when the true independent objects are primes. Integers are abundant configurations; primes are sparse generators. Prime-only summation is the structurally correct

primitive aggregation when you are measuring something intended to track independent contributions rather than configuration proliferation.

2.2 Nonredundant generating families in categories and topoi

In category theory, the analogue of a set of generators is a family of objects that can detect or build the rest of the category. The precise definition varies by context, but the guiding idea is stable: a generator family is a basis-like collection of primitives from which the category's objects and morphisms can be recovered or constructed.

A common notion in a category E is a generating set G such that for any pair of distinct morphisms f and g from X to Y, there exists some generator object G in G and some arrow from G to X that distinguishes f and g after composition. In plain terms: generators are enough to "probe" differences. In many categories of interest, especially Grothendieck topoi, generating families are not only detectors but constructors: every object can be presented as a colimit of generators, often via coproducts and quotients. This is the sense in which generators behave like primes: they are primitives from which composites are assembled.

Nonredundancy enters when we demand minimality. A generating family is nonredundant if no proper subfamily still generates. This is the categorical analogue of irreducibility at the level of the basis itself: you cannot remove one generator without losing the ability to reconstruct the category. In practice, there may be many nonredundant generator choices, just as there are many bases for a vector space. But once a generating family is fixed, it provides a disciplined location for primitive sums and primitive accounting.

For sheaf topoi, there is a particularly canonical route to generators. If a topos is presented as sheaves on a site (C, J), then sheafifications of representables typically form a generating family. From that large generating family one can often extract smaller irredundant subfamilies, depending on the structure of the site and the coverage. In the AI topos construction later, this is how "nonredundant sheaves" arise: they are the chosen generator objects needed to generate all contextual behaviors by gluing and colimits.

2.3 Additivity and why "counting everything" overcounts

The paper's logic rests on a simple but frequently ignored point: whether it is legitimate to "sum over everything" depends on what your sum is supposed to measure. If your target quantity is intended to reflect independent structure, then summing over composites is generally wrong because it counts the same generator content repeatedly in different recombinations.

In arithmetic, this is obvious once you take the generator view seriously. A composite integer such as 12 is not a new independent unit; it is a configuration assembled from 2 and 3. If you are measuring something like "independent generator content," then counting 12 as an

independent contribution is a form of double-counting. Indeed, if you naïvely sum a generator-additive quantity over all integers in some range, you are counting the contribution of prime 2 not only once, but through every even number; prime 3 through every multiple of 3; and so on. The resulting total reflects configuration abundance more than generator novelty.

In a categorical setting, the same pathology appears. If objects are constructed from generators via coproducts, coequalizers, and colimits, then many distinct objects share the same generator content in different glued forms. A functional meant to measure independent degrees of freedom must be compatible with those constructions. In particular, if a functional is additive over coproducts and behaves predictably under quotients and colimits, then it is natural to compute it from generator-level data and reconstruction diagrams, rather than by summing over all objects indiscriminately.

This is the categorical meaning of nonredundancy: the correct primitive accounting lives on the generator family, because that is where independence is located. Composite objects are not irrelevant, but they are not primitive units for the purpose of measuring independent structure. They are derived states whose multiplicity is an emergent consequence of composition and gluing.

2.4 Canonical versus chosen bases: uniqueness and gauge freedom

One reason prime-only summation is so compelling in arithmetic is that primes are canonical: unique factorization gives a privileged basis. In more general settings, including topoi, there is rarely a uniquely determined basis. Multiple generating families can exist, and even multiple irredundant generating families can exist. This raises a foundational question: if the basis is not unique, how can generator-only aggregation be meaningful?

The resolution is to distinguish two levels of statement.

At the representational level, the generator family is a choice, and different choices can lead to different decompositions of objects. This is a kind of gauge freedom: a basis choice changes coordinates in the space of descriptions. At the structural level, meaningful invariants should not depend sensitively on that choice. If a proposed generator-only measure truly tracks something intrinsic about the system, it should be stable under reasonable basis changes or should transform in a controlled way analogous to a change-of-basis transformation.

This is exactly the role played by stability diagnostics later in the paper. If the inferred generator spectrum or novelty curve shifts wildly with minor changes in the site, the coverage, or the generator selection procedure, then the supposed “nonredundant basis” is not tracking independent reality; it is tracking artifacts of representation. Conversely, if generator-level measurements are robust under such changes, then they earn the right to be interpreted as basis-independent content.

Thus, the lack of unique factorization in topoi is not a flaw. It is a feature that forces the framework to be honest. Primes give you canonical factorization; general generator families give you only basis choices. Robustness under basis variation becomes part of the falsifiability criteria for the ontology.

2.5 The irreducible aggregation principle as a categorical template

We can now state the paper's backbone principle in a form that is independent of arithmetic and ready to be instantiated in an AI topos.

Irreducible Aggregation Principle (categorical template): Suppose a category E admits a generating family G, and suppose we care about a functional F that is compatible with the category's construction operations, meaning it behaves additively on coproducts and respects the colimit-based presentations by which objects are built from generators. Then F is determined by its restriction to G together with the gluing relations that assemble composites. In such a setting, primitive aggregation should be performed on G, while composite-level quantities should be reconstructed from generator data rather than counted as independent contributions.

This principle encapsulates the prime-only summation idea without mentioning primes. Primes are simply the clearest archetype: the integers are built freely from primes, so any generator-additive quantity is naturally a prime-level sum, and configuration-level phenomena are derived through multiplicative composition and Euler products. In the topos setting, the generators are "nonredundant sheaves," composites are glued behaviors, and prime-only summation becomes generator-only aggregation. The rest of the paper instantiates this template in an explicit AI topos and then develops measurable novelty and reconstruction diagnostics that can confirm or falsify whether the generator lens is tracking independent structure in real systems.

3. The AI Topos: Contexts, Coverings, and Sheaves of Behavior

The phrase "AI topos" can easily collapse into vague mysticism unless we specify what the objects are, what the morphisms are, and what the sheaf condition is buying us. This section provides a concrete construction that is simultaneously mathematically orthodox and operationally meaningful. The guiding idea is that an AI system does not have one monolithic behavior; it has behavior that varies with context, and the most important structural question is whether these contextual behaviors fit together coherently. A topos of sheaves is a natural home for that question: it formalizes "local descriptions that glue to a global description" and supplies a rigorous notion of what counts as primitive, what counts as derived, and what it means to reconstruct composite behavior from a basis of local evidence. The result is not a

metaphorical topos. It is a bona fide Grothendieck topos built from contexts, coverings, and a sheaf that records model responses in a way that respects restriction and marginalization.

3.1 Contexts as probe-sets and refinement as information order

We begin by defining a site, i.e., a category of contexts. The most practical choice for AI evaluation is a poset category built from probe-sets.

Let P be a universe of probes. A "probe" can be any standardized interrogatory instrument: a prompt template, a task instance, a question, a test scenario, a tool-usage request, a safety challenge, or a sensor slice in non-textual settings. A context is then a finite subset $c \subseteq P$, representing "the set of probes currently under consideration."

We order contexts by refinement: a context d refines c if d contains at least the probes of c. Intuitively, d has more measurement channels, so knowing the model's behavior on d gives you at least as much information as knowing its behavior on c. This yields a poset category C in which there is a morphism d -> c exactly when $d \supseteq c$.

This ordering encodes information flow. Restriction maps will always move from richer contexts to poorer contexts, because you can always forget probes, marginalize, or project away extra components. You cannot generally go the other way without assumptions. This asymmetry is the categorical shadow of a common evaluation fact: it is easier to forget information than to hallucinate missing information reliably.

This context poset becomes the scaffolding on which we hang behavioral data. It also lets us express a crucial distinction: a model may appear consistent at the level of individual probes while failing to be consistent at the level of probe combinations. The topos will be sensitive to exactly this difference, because gluing requires compatibility on overlaps.

3.2 Coverings as "sufficient local evidence" and gluing conditions

A Grothendieck topology on C specifies what families of refinements count as a cover: a set of arrows {c_i -> c} such that "having local data on all the c_i is sufficient to determine the data on c, provided the locals agree on overlaps."

In the probe-set poset, a natural covering rule is:

A family {c_i} covers c if the union of the c_i contains c, and each c_i refines c. In the strictest version, $\bigcup_i c_i = c$. In a more flexible version, $\bigcup_i c_i \supseteq c$, allowing auxiliary probes.

Operationally, this reads as "the context c can be reconstructed from a set of local contexts whose probes collectively span c." The overlaps $c_i \cap c_j$ are the shared probes, the shared measurement coordinates on which locals must agree.

This is where the sheaf condition becomes meaningful for AI. If you collect behavioral summaries on multiple sub-batteries of tests, those summaries should agree when they refer to the same overlapping tests. If they do, then there should exist a coherent global summary for the entire battery. If they do not, the system has a contextual inconsistency: it behaves in a way that cannot be represented as a single global object compatible with all local views.

The topology is a modeling decision, but it is not arbitrary. It encodes the scientific assumption about what counts as "sufficient evidence." In a safety evaluation, you might require stricter covers. In capability evaluation, you might allow auxiliary probes. In systems engineering settings, you might treat certain probes as privileged sensors. Changing the coverage changes what you mean by "gluable," and therefore changes the topos semantics. This is not a flaw; it is the formal way of saying that "global coherence depends on what local views you consider sufficient and authoritative."

3.3 The sheaf of responses: distributions, restrictions, and marginals

Now we define the object of central interest: a sheaf X assigning to each context c a mathematical representation of the model's behavior under the probes in c.

To keep the construction robust, we treat behavior as a distribution rather than a single deterministic output. A probe can have stochastic outcomes (temperature sampling), model variability, tool timing variability, or latent nondeterminism. Even if the model is deterministic, the environment may not be. A distributional representation therefore aligns with how systems are actually tested.

For each context c (a finite set of probes), define X(c) to be the joint response distribution over the probes in c. This can be thought of as the empirical distribution induced by repeated runs, or as a model-estimated distribution if one is accessible. It can also be represented by sufficient statistics rather than full distributions if needed, but the conceptual object is the joint behavior over the probe-set.

If $d \supseteq c$, the restriction map X(d) -> X(c) is marginalization: we forget the probes in $d \setminus c$ and keep only the induced distribution on probes in c. This is the most important structural choice in the entire construction, because it makes the restriction maps concrete, canonical, and grounded.

The sheaf condition then states:

Given a cover $\{c_i \to c\}$, if we have local distributions X(c_i) whose marginals agree on all overlaps $c_i \cap c_j$, then there exists a unique global distribution X(c) whose restrictions to each c_i equal the given locals.

In probabilistic language, this is the classical consistency requirement for a family of marginals: compatible marginals should come from a unique joint distribution. In practice, uniqueness can be weakened to existence or to a constrained family of joints, because empirical estimates are noisy. But even in approximate form, the sheaf condition expresses the intended logic: global behavior is what you get when the local views are compatible.

This gives the topos teeth. It converts "the model seems inconsistent across test batteries" into a precise obstruction: the local response sheaves fail to glue. Conversely, coherence across contexts becomes an internal property of the behavioral object in the topos, not an informal judgment.

3.4 What "global behavior" means in a sheaf semantics

In everyday AI talk, "global behavior" is usually a vague phrase meaning "what the model does in general." In the sheaf semantics, "global behavior" has a sharp meaning: it is the glued object that simultaneously explains all locally observed behaviors and their compatibility across overlaps.

There are three related but distinct senses of "global" that matter here:

Global over a fixed context battery. For a chosen context c (a selected set of probes), the global section X(c) represents the joint behavior on that battery, the coherent integration of all local subsets used in its construction.

Global over the entire probe universe (if one takes a limit). If you enlarge contexts and imagine a directed system of probe-sets whose union approximates all probes you care about, then global behavior can be understood as an inverse limit object: a coherent family of behaviors across all finite contexts. This is a strong notion of globality and may be unattainable in practice, but it is conceptually meaningful.

Global as invariants stable under refinement. Often, what we really mean by "global behavior" is not a single gigantic joint distribution but stable invariants: generator spectra, novelty rates, coupling measures, or reconstruction error curves that remain consistent as contexts change. In the topos setting, such invariants are functionals that respect restrictions and are stable under refinement.

The sheaf semantics helps clarify what is being claimed. If a model is said to have a "global capability," one can ask: does this mean there exists a glued behavioral object on a suitable coverage? Does it mean the behavior is consistent under refinement? Does it mean generator-level invariants stabilize? The topos forces these distinctions into the open.

Most importantly for this paper, "global behavior" is not the same as "the set of all outputs." Global behavior is structure constrained by compatibility, and it is precisely the structure that

can be reconstructed from nonredundant generators if the system admits a generator–configuration decomposition.

3.5 Practical construction from logs, benchmarks, and probes

The construction above sounds abstract, but it can be instantiated in an engineering pipeline. The key is to realize that we do not need to build the entire topos explicitly. We need to build enough of the site and enough of the sheaf to estimate generator structure and test reconstruction.

A practical pipeline has five steps.

First, define the probe universe P. This can be assembled from benchmark items, safety suites, tool-use tasks, domain-specific question banks, and real production logs. The probes must be standardized enough that repeated measurement is meaningful.

Second, define contexts c as probe-sets. In practice, contexts might be mini-batteries: small groups of probes grouped by theme, difficulty, or intended capability. The refinement relation is then simply set inclusion.

Third, choose a coverage J. Operationally, J determines how you decompose a battery into sub-batteries and what overlap consistency you require. A common practical choice is to treat overlaps as shared probes and define covers by set unions. One can also incorporate weighting, importance, or confidence in different probes by adjusting the coverage rules, but the simplest union-based coverage already yields a useful sheaf condition.

Fourth, estimate X(c) for selected contexts. This does not require estimating full joint distributions at huge scale. One can estimate low-order marginals, sufficient statistics, or embedding-based summaries, as long as restriction maps correspond to a well-defined marginalization or projection. The crucial requirement is that overlap agreement be testable.

Fifth, perform gluing and obstruction analysis. Given local estimates, check whether they approximately agree on overlaps. When they do, attempt to reconstruct a global object for the larger context. When they do not, record the inconsistency as a gluing obstruction. These obstructions later serve as signatures of coupling and interaction effects: places where the “independent generator” picture breaks.

Once this infrastructure exists, the prime-only analogy becomes actionable. Nonredundant generators correspond to a minimal family of probes or probe-derived sheaves needed to generate the observed behaviors under gluing. Generator-only novelty can be estimated by tracking which basis elements show genuine distributional updates over time, and configuration-level proliferation can be measured separately and compared. In this way, the AI

topos is not a philosophical flourish; it is a scaffolding for turning “independent structure versus recombination” into a measurable distinction.

4. Nonredundant Sheaves as Independent Degrees of Freedom

The previous section constructed an AI topos as a category of contextual behaviors that satisfy a gluing principle: compatible local response data should assemble into a coherent global behavior. This section identifies where “independence” lives in that topos. The central claim is that independent degrees of freedom correspond to a nonredundant generating family of sheaves. Once such a family is fixed, generator-only aggregation becomes the correct primitive accounting for structure, while all other objects are composites whose abundance reflects recombination, gluing, and quotienting rather than new independent reality. The difficulty, of course, is that unlike arithmetic primes, a topos rarely hands you a unique canonical basis. We therefore specify a disciplined route from canonical generator candidates to an irredundant basis, define redundancy operationally in terms of reconstructibility and conditional dependence, and then propose stability as the primary reality test: if a generator basis is tracking genuine independent structure, it should persist under reasonable changes in contexts, covers, and sampling.

4.1 From representables to generators: how the basis arises

In a Grothendieck topos presented as sheaves on a site (C, J), there is a canonical source of generator candidates: representable presheaves. For each object c in C, the representable presheaf y(c) assigns to any d the set of morphisms from d to c. In the presheaf category, the family {y(c)} generates in a strong sense: it is the universal “coordinate system” for probing objects. When we pass to sheaves, we sheafify these representables, obtaining a(c), and the family {a(c)} typically continues to generate the topos.

In the AI topos constructed earlier, contexts are probe-sets ordered by refinement. The representable associated to a context c embodies “the observable shape” of that context within the site. After sheafification, a(c) becomes the sheaf that faithfully reflects how c is covered and glued under the topology. This matters because the coverage encodes what counts as sufficient local evidence. The generator candidates therefore carry not only the identity of probes but the rules by which those probes are allowed to assemble into global behavior.

This canonical generating family is usually enormous. It is analogous to taking “all primes and all prime powers and all structured prime-combinations” as candidates, when what you really want is a minimal independent set. The core task of this section is to move from “there exists a generator family” to “there exists a nonredundant generator family that deserves to be called the independent degrees of freedom.”

The conceptual translation is:

Contexts produce canonical sheaf generators via representables and sheafification. Nonredundant sheaves are then extracted from these canonical generators by a minimality principle that removes those that can be reconstructed from others under gluing.

4.2 Irredundant generator selection and minimality criteria

A generating family G is nonredundant if no proper subset still generates. In practice, one rarely proves minimality by exhaustive search. Instead, one imposes an explicit criterion of redundancy and uses a selection procedure whose failure modes are visible and testable.

There are three practical minimality criteria that fit the AI topos setting.

Minimal detection. A family G generates if it detects distinctions: whenever two maps differ, some g in G witnesses the difference. Nonredundancy here means that removing any g causes some previously detectable distinction to become invisible. In AI terms, a removed generator eliminates the ability to distinguish two behaviors that were previously distinguishable by probing with that generator.

Minimal construction. A family G generates if every object can be obtained as a colimit of objects from G, possibly followed by quotients. Nonredundancy means that removing any generator breaks the ability to construct at least one behavior object. In AI terms, there exists some composite behavioral pattern that can no longer be reconstructed by gluing and composition from the reduced basis.

Minimal presentation size. Often there is no unique minimal family, but there is a family that minimizes a cost functional, such as total generator "energy," total basis description length, or basis cardinality subject to a reconstruction error constraint. This is the operational approach: select G to minimize basis complexity while keeping reconstruction error below a tolerance. The resulting family is nonredundant relative to the reconstruction criterion.

These criteria are compatible. Minimal detection is closest to the categorical definition, minimal construction aligns with the generator–configuration worldview, and minimal presentation size aligns with empirical estimation. The key is that whichever you choose, you must be explicit about what "generate" means and what constitutes failure.

The selection procedure can be thought of as an irreducible extraction: start with a large candidate family and iteratively remove elements that are redundant according to the chosen criterion, stopping when no further removals are permitted. This yields an irredundant family in the practical sense: every remaining generator contributes something that cannot be recovered from the others under the adopted gluing semantics.

4.3 Redundancy as reconstructibility and conditional dependence

To use nonredundant sheaves as independent degrees of freedom, redundancy must mean more than "similarity." It must mean reconstructibility: the candidate generator contributes no irreducible information because its behavior is already determined by other generators under the system's composition rules.

In the AI topos, where the response sheaf assigns distributions to contexts and restriction maps correspond to marginalization, reconstructibility admits a direct operational translation: a probe or context is redundant if its response distribution is determined, to within tolerance, by the distributions on a chosen basis of other probes or contexts.

This can be formalized using conditional dependence. Consider a candidate generator corresponding to a probe p or a small context c. Let S be a set of basis probes or contexts. Redundancy means that the behavior on p is conditionally determined by the behavior on S. In probabilistic terms, the mutual information between p and the rest of the system collapses once S is known. In modeling terms, there exists a predictive map from S-behavior to p-behavior with low error. In topos terms, p adds no new sheaf-theoretic content because it is a glued consequence of S.

This point is essential: redundancy is not a property of p alone. It is a property of p relative to a chosen set S and relative to a chosen topology J. Change the coverage, and what is reconstructible can change. Change the noise model, and what is predictably determined can change. That is why basis stability will later be treated as a reality test rather than an assumption.

The conditional dependence viewpoint also clarifies the difference between "features" and "generators." A feature can be highly salient and still be redundant if it is determined by other features. A generator is not merely salient; it is irreducible under the adopted composition semantics. When we claim "nonredundant sheaves are independent degrees of freedom," we mean precisely that: they are the smallest set of behavioral primitives from which the rest is reconstructible under gluing.

4.4 Composite sheaves as glued configurations and emergent states

Once a generator basis is fixed, the rest of the topos becomes interpretable as a space of configurations. Composite sheaves arise from generators by the operations that the topos naturally supports: coproducts, colimits, quotients, and gluing along covers. These operations create a vast universe of derived objects, often far larger than the generator family itself. This is the categorical analogue of how a small set of primes generates an enormous set of integers.

In the AI topos, composites correspond to behaviors on larger contexts: joint response distributions on larger probe-sets, or higher-order behavioral objects formed by integrating multiple local behaviors. Even if the generator basis is small, the space of composite contexts can be huge, and the number of distinguishable composite behaviors can grow explosively with context size. This is where the recombination problem lives: the observable surface explodes, even if the independent basis does not.

The generator–configuration distinction suggests a disciplined reading of emergence. Emergent states are not denied; they are located. They are the composites that arise when independent degrees of freedom interact through gluing constraints. In a perfectly independent system, gluing would factorize cleanly, and composite behaviors would be almost entirely reconstructible from generator spectra. In real systems, gluing often introduces interaction terms: incompatibilities, couplings, and higher-order constraints that cannot be reduced to independent generator contributions. These interaction terms are not “new generators” by default; they are the signature of how generators couple.

Thus, the framework separates two kinds of growth:

Basis growth: new independent degrees of freedom appear, requiring expansion of the nonredundant generator family.
Configuration growth: new composites appear through recombination, gluing, and interaction among existing generators.

Counting outputs or composite contexts primarily measures configuration growth. Generator-only aggregation is aimed at basis growth. The distinction is the key to making “novelty” meaningful rather than merely combinatorial.

4.5 Stability of generator bases as a reality test

Because generator bases are rarely unique in topoi, any claim that “these are the independent degrees of freedom” must be validated by stability. Stability is the categorical counterpart of “coordinate invariance”: if independence is real, it should not depend strongly on superficial choices in how we slice the system.

We therefore treat basis stability as the primary reality test. The test has multiple axes:

Stability under resampling. If you rebuild the basis from different subsets of data, you should recover essentially the same generator family or at least a family that differs only by near-equivalences.

Stability under windowing. If you compute the basis on different time windows, the basis should persist unless the system is genuinely changing. Large swings without corresponding behavioral shifts signal that the basis is an artifact.

Stability under topology variation. If you slightly change the covering rule or allowable auxiliary probes, the basis should not collapse. Some drift is expected, but catastrophic non-robustness indicates that "generator" is being defined by the coverage rather than discovered in the system.

Stability under representation changes. If you change the response representation (full distributions versus sufficient statistics, different embedding summaries), basis-level conclusions should remain consistent if they track intrinsic structure.

When stability holds, the basis earns interpretive authority: it becomes reasonable to speak of nonredundant sheaves as independent degrees of freedom. When stability fails, that failure is informative: it indicates that the system does not admit a clean generator–configuration separation under the chosen semantics, or that the measurement apparatus is insufficient.

This is the conceptual hinge of the paper. The framework does not demand that AI behavior obey a prime-like ontology. It demands that if one claims such an ontology, it must show robustness, reconstructibility, and null-model separation. Nonredundant sheaves become "independent degrees of freedom" only to the extent that they are stable and predictive. In this way, the prime-only summation idea is not imported as dogma. It is imported as a falsifiable proposal for what should be counted once.

5. Prime-Only Summation Translated: Generator-Only Aggregation Functionals

This section is the hinge where the prime-only summation idea becomes an explicit measurement rule inside the AI topos. The shift is subtle but decisive. In arithmetic, restricting a sum to primes is justified because primes are the irreducible generators of multiplicative composition and because a large class of quantities of interest are additive over prime exponents or become additive after taking logarithms of multiplicative generating functions. The analogue in a topos is to restrict primitive aggregation to a nonredundant generating family of sheaves and to require that the quantities being aggregated respect the constructions by which composite objects are formed. If we do this, we obtain a principled class of generator-only functionals that measure independent degrees of freedom and resist inflation by recombination. If we do not do this, configuration abundance overwhelms the signal and "novelty" becomes almost guaranteed, regardless of whether independent structure is actually growing.

5.1 Defining generator-only sums for categorical invariants

Fix an AI topos E built from contexts and coverings, and fix a nonredundant generating family G of sheaves, interpreted as independent degrees of freedom. A generator-only aggregation

functional begins with a local assignment on generators and then extends it to composite objects via the category's construction rules.

The simplest form looks like this. Define a weight function w on generators, w: G -> R_{\ge 0}. Then define the generator-only sum

S(G) = sum over g in G of w(g).

This looks trivial, but it becomes meaningful when w(g) is not arbitrary: it is chosen to measure an intrinsic contribution of generator g to the system's behavior, and it is chosen so that composites do not create new primitive contributions unless they introduce genuinely new generators or genuinely new coupling structure.

More generally, we can define a generator-only functional F on objects X of E by specifying:

A generator score function s(g; X) that measures how much generator g is present, activated, or required to describe X.
A weighting w(g) that measures how expensive, rare, or important g is.
Then

F(X) = sum over g in G of w(g) * s(g; X).

This is the true analogue of prime-only sums with exponents: primes correspond to generators; prime exponents correspond to occupancy, activation, or necessity scores; and the sum is the irreducible accounting of generator contributions. The content is not in the symbol "sum," but in the discipline that s(g; X) must be defined so that it respects the topos's notion of composition and gluing rather than arbitrarily rewarding configuration-level complexity.

There is an important conceptual separation here. The generator family G is a structural choice that defines what "independent" means. The function s(g; X) is the mechanism by which a particular object X draws on that independent basis. Together they turn the prime-only idea into a basis-only accounting rule: measure how much X uses the basis, not how large or diverse X appears at the surface.

5.2 Additive and colimit-respecting functionals as admissible measures

Prime-only summation is canonical in arithmetic because it aligns with additivity on generators. In the topos setting, the analogue is to restrict attention to functionals that respect the ways objects are built from generators: coproducts, colimits, quotients, and gluing along covers. These are the admissible measures for which generator-only aggregation is not merely a preference but a structural necessity.

A functional F is admissible in this sense if it satisfies two broad properties.

First, it should be additive on coproducts. If X and Y are disjoint components, the measure of their union should be the sum of measures. This is the categorical version of additivity over independent modes. Without this, “basis content” becomes hard to interpret.

Second, it should respect the presentations of objects by generators. In a Grothendieck topos, many objects can be described as coequalizers of maps between coproducts of generators. If F respects these colimits, then F(X) can be computed from generator-level data and the gluing relations. This is the mechanism by which “everything is determined by the basis” becomes a theorem-shaped statement rather than a slogan.

Operationally, the admissibility requirement means that the measure is not allowed to treat every composite object as a new primitive contribution. It must be compatible with the category’s composition semantics. If the topos says that object X is built by gluing local pieces with specified overlap consistency, then F should compute X’s contribution by looking at the pieces and how they glue, not by treating X as an independent atom merely because it is a distinct object.

This admissibility condition is the mathematical firewall against novelty inflation. It forces evaluation to follow the structure of composition rather than the seduction of surface variety.

5.3 Overcounting corrections: why composites must not be primitive

The recombination problem in AI evaluation is, at heart, an overcounting problem. The topos language provides a precise way to state what is being overcounted: the same generator contributions are being counted repeatedly across many glued configurations.

Suppose we measure novelty by counting distinct composite contexts or distinct outputs. Even if the generator basis is fixed and small, the number of composite contexts grows combinatorially, and the number of distinct outputs grows astronomically. A naive novelty count will therefore increase indefinitely. This is not a deep empirical discovery; it is an inevitability of configuration spaces.

Generator-only aggregation corrects this by imposing a strict accounting rule: composites do not earn primitive credit unless they require new generators or reveal new irreducible coupling structure. Everything else is treated as derived.

There are two canonical correction mechanisms, mirroring what happens in arithmetic.

Generator projection. Map each composite object X to a generator-content vector, describing which generators it uses and how. Then define novelty or complexity as a function of that vector, not as a function of the raw object identity. Many distinct composites will map to the same or similar generator vectors, collapsing configuration multiplicity into a smaller space of independent structure.

Log-linearization. In arithmetic, multiplicative configuration counts become additive prime sums after applying logarithms to generating functions. In the topos setting, many configuration-level totals become generator-level sums after applying an appropriate linearization that converts gluing and composition into additive contributions plus interaction terms. The log of a partition-like functional is the cleanest example, but the deeper principle is that you should use a transform that exposes generator-only content and isolates coupling as a correction term.

Both mechanisms enforce the same moral: treat composites as reconstructible. Do not let them pretend to be independent just because they are numerous.

This is also where falsifiability enters. If overcounting corrections systematically fail to reduce apparent novelty, that failure is evidence that the system's behavior is dominated by interaction and coupling rather than by recombination over a stable basis. In that case, the generator ontology may be inadequate, and the failure is not embarrassing; it is informative.

5.4 Multiple weightings: scale, impact, rarity, and intervention cost

Prime-only summation becomes interesting because different weights on primes reveal different structural facts. The same is true for generator-only sums. A basis-only approach does not commit you to a single notion of "importance." It provides a framework in which multiple weightings correspond to multiple legitimate scientific questions.

Scale weighting measures how much independent structure exists at different granularity levels. A generator corresponding to a broad capability might be weighted differently than a generator corresponding to a narrow micro-feature. This is how one prevents a basis from degenerating into a grab-bag of tiny fragments that inflate basis size without increasing explanatory power.

Impact weighting measures significance. Some independent degrees of freedom matter more for safety, reliability, or capability. Weighting by impact allows generator-only novelty to serve as an audit tool: it tracks not merely how much novelty exists but where it matters.

Rarity weighting treats independent degrees of freedom as events with varying prevalence. If a generator is rarely activated or rarely discovered, it may deserve higher weight, analogous to information-theoretic surprise. This naturally connects to generator energies defined as negative logs of activation rates.

Intervention-cost weighting measures how hard it is to elicit, control, or suppress a generator. This weighting is particularly relevant for systems engineering and safety: an "expensive" generator might represent a brittle or hard-to-control mode, and its presence may matter more than its frequency.

These weightings can coexist. The framework's discipline is not that it produces one number; it is that it produces a coherent family of numbers, each grounded in generator-only accounting

and resistant to configuration inflation. The weights become the knobs by which the evaluator states what they care about.

5.5 Interpreting “prime-only” as “basis-only” without metaphor

At this point we can state the translation cleanly, with no reliance on poetic analogy.

Prime-only summation is a special case of a general rule: when a system admits a generator–configuration decomposition and when the quantity of interest is additive over generators or becomes additive after an appropriate linearization, the correct primitive aggregation is over generators, not configurations.

In arithmetic, the generators are primes and the configurations are integers. In the AI topos, the generators are nonredundant sheaves and the configurations are glued behaviors over contexts. The prime-only restriction corresponds to restricting primitive sums to the generator family G. The move is justified not by aesthetic preference but by structural necessity: summing over all composites mismeasures independence by overcounting derived states.

To say “prime-only is basis-only” is therefore to say something precise:

The independent degrees of freedom are those objects that must be included in an irredundant generating family, and the correct primitive evaluation functionals are those that respect the category’s composition operations and thereby compute composite contributions from generator data plus explicit coupling corrections.

This is the mathematical backbone for the rest of the paper’s diagnostic program. Once generator-only functionals are defined, we can propose testable claims about basis stability, novelty thinning, reconstruction accuracy, and coupling localization. The prime-only summation idea does not remain an analogy; it becomes a disciplined accounting rule that can succeed, fail, and be measured in real AI systems.

6. Euler-Product Analogues: From Configuration Totals to Generator Spectra

The prime-only summation idea becomes most powerful in number theory when it is paired with Euler products. The Euler product is not merely a curiosity; it is the algebraic mechanism by which a vast configuration sum over integers factorizes into independent contributions from primes. Once factorization exists, taking the logarithm turns multiplicative structure into additive structure, converting configuration accounting into generator accounting. This section transports that mechanism into the AI topos setting. We define configuration spaces as multisets of nonredundant generators, assign generator energies that induce additive costs on configurations, construct partition-like functionals whose factorization expresses independence,

and then show how log-linearization exposes generator-only structure. Finally, we address the essential realism: factorization is rarely exact in real AI systems. When it fails, it fails in structured ways, and those structured failures become a measurable signature of coupling and emergence.

6.1 Configuration spaces as multisets of generators

Fix a nonredundant generating family G in the AI topos E. Intuitively, each g in G represents an independent degree of freedom, and composite behaviors are built by assembling these degrees of freedom under the topos's composition rules. To obtain an Euler-product analogue, we begin with an idealized configuration model that mirrors the arithmetic situation.

In arithmetic, an integer corresponds to a finite multiset of primes with multiplicity. The topos analogue is to model a configuration as a finite multiset of generators. Concretely, define an occupancy function k on G, where k(g) is a nonnegative integer and only finitely many k(g) are nonzero. The multiset described by k is then the formal composite object made by taking k(g) copies of each generator g and assembling them under a commutative "bag" operation.

In categorical language, the simplest "bag" operation is the coproduct. A configuration corresponding to k is a coproduct of generator copies: a disjoint union of primitive components. This is not yet the full richness of gluing in a topos, but it isolates the part of composition that is closest to "independent occupancy." It is the correct starting point because Euler-product factorization is, at root, the algebra of independent occupancy counts.

Operationally, this configuration model is a disciplined abstraction of what AI evaluators often do informally. They treat a complex behavior as composed of multiple primitive subskills or micro-features. Representing configurations as multisets forces that intuition into a formal object: a configuration is not a mysterious emergent whole, but a counted assembly of basis elements.

This is also where overcounting becomes visible. The number of distinct multisets grows rapidly as you allow larger total occupancy. Even if the generator family is fixed, configuration space explodes. Therefore, any naive "sum over configurations" is sensitive to combinatorial multiplicity. The Euler-product machinery exists precisely to let us handle such configuration sums without confusing multiplicity with independence.

6.2 Generator energies and additive costs in the AI topos

Euler products become meaningful once we assign each generator a scalar that acts like an energy or cost. In arithmetic thermodynamics, assigning e(p) = log p is the natural move because it links multiplicative composition to additive cost, and it makes zeta-like partition

functions appear. In the AI topos, the analogue is to assign each generator g in G an energy e(g) that quantifies how costly, rare, or significant that generator is.

The defining property is additivity on configurations. If a configuration has occupancy k(g) for each generator g, its total energy is

E(k) = sum over g in G of k(g) * e(g).

This is the topos analogue of additive cost on prime exponents. It encodes the claim that in the independence regime, the total “cost” of a composite behavior is just the sum of its independent parts, counted with multiplicity.

The interpretation of e(g) depends on what you intend to measure. It can represent description length, rarity, difficulty of elicitation, intervention cost, or any other scalar that is meant to be primitive at the generator level. The framework does not force a single semantics, but it forces a single discipline: whatever semantics you choose, the composite cost must be induced additively from generator costs if you want an Euler-product-like factorization to be appropriate.

This energy assignment also creates a generator spectrum. The system’s independent structure is no longer described solely by the set G; it is described by the spectrum of energies across G. In arithmetic, prime sizes encode a distribution that shapes the behavior of sums and products. In the AI topos, the energy spectrum shapes which generators dominate, which are rare, and how novelty and reconstruction behave as observational or computational parameters change.

6.3 Partition-like functionals and factorization under independence

Once we have a configuration space and an energy function, we can define partition-like functionals. A partition function is simply a weighted sum over configurations:

Z(beta) = sum over finite multisets k of exp(-beta * E(k)),

where beta is a control parameter. The control parameter can be interpreted as inverse temperature, a budget constraint, a regularization strength, or more generally as the knob that changes how strongly the system prefers low-energy versus high-energy configurations.

In the pure independence regime, the crucial fact is that configurations decompose into independent occupancy counts for each generator. That independence causes factorization. For each generator g, the possible occupancies contribute a geometric series:

sum over k(g) >= 0 of exp(-beta * k(g) * e(g)) = 1 / (1 - exp(-beta e(g))),

provided exp(-beta e(g)) < 1.

Since generators are independent, the total partition function factorizes as a product over g:

Z(beta) = product over g in G of 1 / (1 - exp(-beta e(g))).

This is the Euler-product analogue. Instead of “product over primes,” it is “product over generators.” Instead of “integers built from primes,” it is “configurations built from generators.” The meaning of the product is the same: configuration accounting decomposes into independent generator contributions when the system admits a bag-of-generators structure.

This factorization is not merely pretty. It provides a practical computational advantage. It collapses a combinatorial sum over an enormous space into a product over a relatively small generator set. It also provides a conceptual advantage: it isolates the generator spectrum as the primitive object controlling global configuration statistics.

Most importantly, the factorization exposes a falsifiable statement. If the topos behavior is truly generated by approximately independent degrees of freedom, then certain empirical configuration statistics should be predictable from a learned generator energy spectrum. If those predictions fail systematically, the independence premise is wrong or incomplete, and the failure can be measured.

6.4 Log-linearization: why logs expose generator-only structure

In number theory, Euler products become most useful after taking logarithms. The log converts products into sums, and it converts the configuration-level multiplicativity into generator-level additivity. The same is true here.

Taking logs yields:

log Z(beta) = sum over g in G of -log(1 - exp(-beta e(g))).

This expression is generator-only. It contains no explicit configuration enumeration. It is a sum over the generator spectrum. This is the precise sense in which log-linearization “exposes” the generator structure: it strips away combinatorial multiplicity and reveals the additive contributions from irreducibles.

This move has two deeper interpretations that matter for AI.

First, it is an extraction of primitive degrees of freedom. The log of a configuration functional counts how many “independent modes” contribute, weighted by their energy. It is therefore a natural candidate for measuring independent capacity, independent novelty potential, or independent failure-mode complexity.

Second, it isolates the correct place for emergence. If independence holds, the generator-only sum fully explains the configuration statistics. If independence fails, the deviations from the log-additive form represent structured coupling, not random noise. Thus, the log does not only reveal the basis; it provides the baseline against which interactions can be measured.

In practice, log-linearization is also computationally attractive. Many empirical measures are multiplicative or combinatorial in the raw representation but become additive after a log transform. The paper later leverages this to define generator-only novelty and to separate basis growth from configuration growth.

6.5 Interaction terms as controlled violations of factorization

No serious AI system is expected to behave as a perfectly factorized bag-of-generators. Contexts interact, features couple, and behaviors depend on joint constraints that cannot be reduced to independent occupancy counts. The value of the Euler-product analogue is not that it pretends interactions do not exist. The value is that it gives a clean independence baseline and a structured language for measuring how interactions violate it.

There are two primary ways factorization fails, corresponding to two kinds of coupling.

Gluing obstructions. In the topos, local behaviors may fail to glue into a global behavior because overlap consistency fails. This is a structural interaction: the joint behavior is not simply the product of independent locals because the locals constrain one another in incompatible ways. Empirically, this appears as marginal inconsistency: locally estimated distributions do not admit a single joint distribution that matches all of them simultaneously.

Energy non-additivity. Even if gluing is consistent, the effective cost of composing generators may not be additive. The joint behavior might have synergies or penalties. Empirically, this appears as systematic deviations from predicted configuration frequencies based on additive generator energies.

Both failures can be modeled as interaction corrections to the factorized log form. Instead of

log Z(beta) = sum over g of base(g),

we get

log Z(beta) = sum over g of base(g) + sum over pairs (g,h) of coupling(g,h) + higher-order terms.

The point is not to insist on any particular expansion, but to insist that interactions occupy explicit correction terms rather than being smeared into the definition of the basis. This keeps the ontology clean. Generators represent the candidate independent degrees of freedom; couplings represent how those degrees of freedom fail to be independent in the real system.

This sets up a sharp empirical program. One first learns a generator spectrum e(g) that best explains observed configuration statistics under the factorized model. Then one measures residuals and asks whether they are localized and structured, as coupling theory would suggest, or whether they are diffuse and unrecoverable, which would indicate that the generator picture itself is not appropriate. A system whose behavior is mostly recombination over a stable basis

should exhibit factorization as a good first approximation with relatively small, interpretable coupling corrections. A system whose behavior is fundamentally entangled across contexts should resist this decomposition, and that resistance is itself a falsifiable outcome.

In this way, Euler-product analogues become more than an aesthetic bridge from number theory. They become a practical measurement architecture: a way to map configuration-level totals to generator spectra, to use log-linearization as the basis-extraction operator, and to treat interaction terms as the measurable signature of emergence rather than as an excuse for unprincipled counting.

7. Novelty and Surprise: Measuring Irreducible Growth in AI Systems

If the generator–configuration distinction is correct, then the central empirical task becomes clear: we need a novelty measure that tracks *basis growth* rather than *configuration proliferation*. In ordinary evaluation practice, novelty is often confused with diversity: new outputs, new chains of reasoning, new tool-use sequences, new "styles," new problem instances, new prompt variants. But diversity can grow indefinitely even if the underlying system is only recombining a fixed set of primitives. The framework of this paper demands a stricter notion: novelty should be credited only when the system reveals new irreducible generator content or when it reveals new structured coupling among existing generators that cannot be reduced to previously known gluing behavior. This section makes that demand operational. We define generator activation as a measurable event derived from observational updates, build a topos-native novelty functional designed to be robust in finite samples, explain how it separates novelty from diversity, and propose predicted regimes of novelty growth. We then show how coupling diagnostics distinguish "new basis elements" from "new interactions," preserving the interpretive discipline of the prime-only analogy while still accommodating genuine emergent behavior.

7.1 Generator activation and novelty events from observational updates

Fix a nonredundant generating family G in the AI topos. Each generator g corresponds to an irreducible behavioral primitive: a minimal sheaf component that cannot be reconstructed from others under the adopted coverage and semantics. Generator activation is the event that an observation provides new information about g in a way that cannot be explained as mere resampling noise.

To define this, we separate two layers: a representation of behavior at g, and a method for deciding when it has changed in a meaningful way.

Representation. For each generator g and each observational window t (a batch of runs, a time segment, a dataset slice), we maintain an empirical or estimated behavioral summary B_t(g). In the distributional sheaf construction, B_t(g) can be the response distribution on the probe-set underlying g. If g is a context generator rather than a singleton probe, B_t(g) is the joint behavior on that context. In more compressed settings, B_t(g) could be a vector of sufficient statistics: accuracy on a micro-suite, a calibration curve, an embedding distribution, or any stable behavioral signature that respects restriction/marginalization.

Update detection. A novelty event should not trigger merely because B_t(g) is different from B_{t-1}(g); it should trigger because the difference exceeds what would be expected under noise, sampling variability, or the null model. The simplest disciplined approach is to compute a divergence or distance D(B_t(g), B_{t-1}(g)) and declare an activation if it crosses a threshold that is calibrated by repeated trials or bootstrap estimates. The threshold can be adaptive: tighter for well-sampled generators, looser for sparse ones.

This definition yields a boolean activation indicator I_t(g), equal to 1 when generator g exhibits a novelty event in window t and 0 otherwise. Importantly, activation is generator-relative, not output-relative. A single observation can activate multiple generators if it changes multiple irreducible components. Conversely, a thousand diverse outputs might activate none if they are all explainable as recombinations within the existing basis.

The conceptual payoff is immediate: novelty becomes a sparse set of generator events rather than an ever-growing list of distinct outputs. This is exactly the “prime-only” discipline translated: we count changes in irreducibles, not changes in composites.

7.2 A topos-native novelty functional and finite-sample estimators

With activation events defined, we construct a novelty functional that is native to the generator–configuration semantics of the topos. The functional must satisfy three requirements.

First, it should accumulate novelty at the generator level and avoid configuration inflation. Second, it should incorporate diminishing returns: the first time a generator is activated is more informative than the hundredth time. Third, it should behave sensibly under finite samples, where noisy updates and sparse observations can create false novelty.

A simple functional that meets these requirements is:

N(T) = sum over g in G of w(g) * h(C_T(g)),

where C_T(g) = sum_{t=1..T} I_t(g) is the count of novelty events for generator g up to time T, w(g) is a weight capturing importance or energy, and h is a concave saturating function. A common choice is h(n) = 1 - exp(-alpha n), which rises quickly at first and then saturates, but any

monotone concave function works. The point is that repeated activations of the same generator should not create linear novelty growth, because that would again reward recombination or repeated rediscovery.

Finite-sample behavior matters. In sparse settings, noise can trigger false activations, inflating novelty. The estimator therefore needs guardrails:

A baseline model of expected variation. For each generator g, estimate the distribution of D(B_t(g), B_{t-1}(g)) under repeated sampling in a stationary regime. Use this to set thresholds that maintain a fixed false activation rate.

A persistence rule. Declare novelty only if the change persists for k consecutive windows, or if the posterior probability of a shift exceeds a confidence level. This prevents single-window blips from being counted.

A null-model correction. Run a permutation or label-scramble baseline that breaks meaningful structure while preserving sampling statistics. The novelty functional computed on the scrambled data provides an expected inflation floor. True generator novelty should exceed this floor by a wide margin.

These guardrails are not optional details. They are what makes the novelty functional testable. Without them, generator novelty collapses into a subjective labeling exercise. With them, generator novelty becomes a statistically controlled measure of irreducible change.

7.3 Distinguishing novelty from diversity in outputs and trajectories

A key claim of this paper is that diversity and novelty are different kinds of growth. Diversity is growth in the configuration space: new strings, new tool sequences, new reasoning paths, new prompt variants. Novelty is growth in the generator basis or in its coupling structure. A mature evaluation framework must measure both, but it must not confuse one for the other.

The generator novelty functional N(T) is intentionally insensitive to sheer output diversity. It can remain flat even as outputs become wildly varied, provided the generator basis and its coupling structure remain unchanged. Conversely, it can spike even when surface diversity appears modest, if a new irreducible generator becomes active or if existing generators reorganize in a way that cannot be reconstructed from prior structure.

This separation leads to a new diagnostic pair:

Configuration diversity metrics: counts of distinct outputs, entropy of outputs, variety of trajectories, distribution of tool calls, etc. These will almost always grow with sampling.

Generator novelty metrics: N(T) and related generator spectra shifts. These may plateau or thin.

The empirical signature of recombination-dominant systems is a widening gap: diversity grows rapidly, novelty grows slowly. The empirical signature of basis-expanding systems is that novelty continues to grow in a stable way, and that this growth correlates with improved reconstruction performance and consistent shifts in generator spectra.

This is the most direct way to operationalize the original concern: if we want to know whether an AI is genuinely acquiring new independent structure, we should track generator novelty, not output variety.

7.4 Predicted regimes: early rapid discovery and long tail thinning

The prime-only analogy suggests a qualitative shape for novelty growth that is both plausible and testable. Primes are sparse among integers, and discovering “new primes” becomes rarer as numbers grow. Similarly, if independent degrees of freedom exist as a sparse basis underlying a vast configuration space, then new generator discoveries should thin as the system matures.

This yields a predicted regime structure for generator novelty over time or over increasing observational budget.

Early rapid discovery. At the beginning of exploration, many generators have not yet been activated. Novelty events occur frequently because the evaluator is quickly finding irreducible modes that were not previously measured or characterized. N(T) rises quickly.

Transition regime. After a moderate observational budget, most high-prevalence generators have been activated. Novelty events become rarer, and updates tend to refine estimates of known generators rather than reveal new ones. N(T) continues to rise but with decreasing slope.

Long tail thinning. Eventually, only rare or high-energy generators remain undiscovered. Activation events become infrequent. Novelty growth slows dramatically and may approach a plateau under a fixed probing regime.

This prediction is not merely descriptive. It is testable through controlled probing. If you expand the probe universe, change the coverage, or increase the diversity of contexts, you may force the discovery of new generators, causing new bursts. But under a stable probe regime, thinning is expected if the system’s basis is not expanding.

Moreover, the shape of thinning carries information about the generator energy spectrum. If energy is defined as negative log activation probability, then the novelty curve is effectively sampling the tail of that spectrum. A system with many rare independent modes will show a different thinning profile than a system with a small basis.

Thus, novelty curves are not just scores; they are empirical windows into the structure of independent degrees of freedom.

7.5 Coupling diagnostics: when novelty is interaction-driven

A danger arises precisely when we become disciplined about basis growth: some genuinely new phenomena may not correspond to new generators at all. They may correspond to new couplings among existing generators, or to new gluing constraints that alter how locals assemble into globals. If we treat all novelty as basis expansion, we will misclassify interaction-driven emergence.

The framework therefore includes coupling diagnostics as a parallel track. The basic idea is simple: if a behavior shift appears in composite contexts but can be explained by new interaction terms without introducing new generators, then the novelty is coupling-driven rather than generator-driven.

Operationally, coupling is detected by failures of factorization and failures of gluing.

Factorization residuals. In the Euler-product analogue, independence predicts that configuration statistics factorize into generator contributions. Fit the best generator spectrum under the factorized model, then measure residuals on composite contexts. If residuals are localized and structured, they represent coupling. If residuals are diffuse, the generator model is inadequate.

Gluing obstructions. In the sheaf semantics, local behaviors on a cover should glue when they agree on overlaps. When observed locals fail to glue despite overlap agreement within tolerance, that failure indicates a higher-order inconsistency: the existence of constraints that are not captured by pairwise overlap consistency. Such obstructions are a natural location for emergent coupling.

Interaction novelty functional. One can define a parallel novelty measure that accumulates changes in coupling terms rather than in generators. For example, define a set of monitored couplings between generator pairs or small subsets, and track when the coupling estimates shift significantly. This yields an interaction novelty curve that can rise even when generator novelty plateaus.

The combined picture is the strongest form of the paper's claim. Independent growth in AI systems can occur in two ways:

New generators appear, increasing basis size.
New couplings appear or reorganize, increasing interaction complexity.

A disciplined novelty measure must separate these. Generator-only novelty is the prime-only analogue. Coupling novelty is the controlled correction term. When both are tracked, we can

say something precise about what is “new” in the system: new independent modes, new interaction structure, or merely new configurations assembled from old parts.

This section therefore establishes the measurement heart of the framework. Novelty is recast as a generator-level and coupling-level accounting problem, equipped with finite-sample estimators and falsifiable predicted regimes. The next section builds on these measurements to develop model selection, diagnostics, and explicit falsifiers for the generator ontology itself.

8. Model Selection and Diagnostics: Falsifying the Generator Ontology

A generator ontology is only as good as its ability to be wrong in public. The temptation in any foundational framework is to treat failures as “complexity” and successes as “confirmation.” This paper rejects that posture. The generator–configuration decomposition is proposed as a testable hypothesis about how AI behavior is organized, not as an interpretive style. The point of this section is to define competing hypotheses that can plausibly explain observed behavior without invoking irreducible generators, and then to supply diagnostics that can select among these hypotheses with explicit falsifiers. The guiding principle is: if generator-only aggregation is real, it should produce stable bases, predictable spectra, and reconstructible composite behavior, and it should outperform null models that destroy compositional structure. If it does not, the ontology is not merely unproven; it is disconfirmed in the sense relevant to scientific modeling.

8.1 Competing hypotheses: additive structure, random labeling, mixed regimes

The generator ontology sits between two extremes. On one side is a purely additive view: behavior decomposes into independent primitives, and composites are almost entirely reconstructible. On the other side is a purely configuration view: behavior is effectively a high-dimensional entangled object where no stable irreducible basis exists and where “primitives” are artifacts of the evaluator’s coordinate system. Real AI systems may occupy mixed regimes between these.

We therefore define three competing hypotheses.

Additive-generator hypothesis. There exists a relatively stable irredundant generating family G such that a large fraction of observed composite behavior can be reconstructed from generator-level summaries plus a limited set of coupling corrections. Under this hypothesis, generator spectra stabilize, novelty thins, and reconstruction error declines as the basis is learned.

Random-labeling hypothesis. Apparent generator structure is largely an artifact of how probes are named, grouped, or represented. Any “basis” extracted from data is unstable under small perturbations, and generator-level statistics are not significantly better than what would be

obtained by randomizing probe labels or scrambling measurement coordinates. Under this hypothesis, basis stability fails and null models match or exceed the explanatory power of the generator approach.

Mixed-regime hypothesis. Some parts of behavior admit generator structure, while others are dominated by coupling and entanglement. Under this hypothesis, a generator basis may exist but only locally: in restricted domains, limited context sizes, or particular probe families. Reconstruction works in some sectors and fails in others, and coupling terms are not merely small corrections but dominant in specific regions of the context space.

The goal is not to proclaim one hypothesis true in the abstract, but to determine which regime best describes a given AI system under a given site of contexts and coverage. Importantly, the framework permits the mixed-regime outcome as a legitimate scientific result. A partial generator ontology, valid only in specific context layers, is still informative and still supports principled measurement in those layers.

8.2 Diagnostic plots: basis stability, spectra, and reconstruction error

Diagnostics should be simple enough to interpret and hard enough to fake. The most useful diagnostics in this framework fall into three families: stability diagnostics, spectral diagnostics, and reconstruction diagnostics.

Basis stability curves. Construct generator bases under multiple resamplings of the data, multiple time windows, or multiple small variations of the coverage. Then quantify overlap between bases. Plot overlap as a function of sample size and perturbation strength. Under the additive-generator hypothesis, stability should increase with sample size and remain high under small perturbations. Under the random-labeling hypothesis, stability remains low and does not improve meaningfully with sample size. Under mixed regimes, stability is high within a sector but collapses when contexts cross sector boundaries.

Generator spectrum plots. Once a basis is chosen, estimate generator energies or weights (rarity, cost, surprise). Plot the distribution of energies, and plot how it evolves with time or observational budget. Under a genuine generator ontology, spectra should converge: early in measurement, energies fluctuate, but they stabilize as sampling improves. Under random-labeling, spectra may appear to stabilize but will be highly sensitive to representation choices and will not predict composite statistics out of sample.

Reconstruction error curves. Choose composite contexts and attempt to reconstruct their behavior from generator-level data using the gluing and factorization model. Plot reconstruction error as a function of basis size, as a function of context size, and as a function of observational budget. Under generator structure, increasing basis size should reduce error systematically until it reaches a coupling-dominated floor. Under random-labeling, error will not decrease reliably

and may not beat simple baselines. Under mixed regimes, error decreases in some sectors and explodes in others, revealing the boundaries of generator validity.

These plots are powerful because they separate three questions: does a basis exist, does it have a stable spectrum, and does it actually reconstruct what it claims to generate? Together they define a coherent model selection toolkit.

8.3 Robustness to noise, missingness, and observational bias

Any evaluation framework that relies on learned bases and estimated distributions must confront the realities of incomplete and biased observation. AI systems are not observed uniformly. Benchmarks overrepresent some capabilities and underrepresent others. Production logs are shaped by user populations, safety filters, and platform constraints. Missingness can be systematic, not random. Without explicit robustness procedures, generator extraction becomes a mirror of the dataset rather than a discovery of the system.

Robustness here has three components.

Noise tolerance. Activation events and basis shifts must be calibrated against sampling variability. This requires per-generator uncertainty estimates and thresholding that adapts to sample size. Diagnostics should be reported with confidence bands or bootstrap ranges, so basis changes are not mistaken for noise.

Missingness handling. Some probes are rarely observed. If basis selection is purely frequency-driven, the basis will be biased toward what is common rather than what is irreducible. The framework therefore encourages energy-based weighting that can upweight rare but important generators, and it encourages design of targeted probing sweeps to fill observational gaps.

Bias correction. If observation is skewed by user populations or by safety gating, then generator estimates will reflect the pipeline rather than the model. A practical remedy is to treat different observational sources as distinct context families and to test whether generator bases inferred from each source glue coherently. If they do not, the mismatch is itself a diagnostic: it identifies where observational bias is altering the apparent structure.

Robustness is not an add-on. It is part of falsifiability. A generator ontology that only holds under one highly curated dataset but collapses under modest changes in sampling is not measuring independent reality. It is measuring a particular measurement regime.

8.4 Null models: permutation baselines and factor-scrambling analogues

Null models provide the most reliable reality check because they answer the question: could we have obtained similar signatures without meaningful structure?

The simplest null is label permutation. Randomly permute probe labels while preserving the marginal distribution of outcomes. If generator-only novelty and spectra remain similar after permutation, the supposed generator structure is an artifact of marginal statistics, not of compositional organization.

A stronger null is factor scrambling. In arithmetic, if you scramble prime factorizations, Euler-product structure collapses while preserving some superficial counts. The analogue here is to scramble the mapping between contexts and their overlaps, breaking the gluing geometry while preserving local statistics. For example, keep the distributional summaries for each context but randomly reassign which overlaps are shared. This destroys real overlap consistency while preserving marginal complexity. A genuine sheaf-based structure should fail to glue under this null, and generator reconstruction should degrade sharply.

Another null is independence forcing. Replace joint context distributions with products of their marginals, thereby enforcing factorization. If the generator model cannot outperform this forced-independence baseline, then the claimed generator basis is not doing meaningful work. Conversely, if the generator model significantly outperforms it, the difference quantifies the importance of coupling.

Finally, there is a temporal null. Shuffle the time ordering of observations. If novelty curves remain similar after shuffling, then novelty is not tracking growth or change; it is tracking sampling fluctuations.

Null models should be treated as first-class citizens of the framework. The core claim of generator ontology is not that it can fit data; almost anything can fit data. The claim is that it captures meaningful compositional structure. Nulls are what test that claim.

8.5 Clear falsifiers: instability, nonreconstructibility, and null equivalence

To treat the generator ontology as scientific, we must specify what would falsify it. The falsifiers below are designed to be decisive rather than interpretive.

Instability. If inferred generator bases vary wildly across resamples, windows, or small coverage changes, and if this instability does not diminish with increased data, then the basis is not real in the intended sense. It may be a coordinate artifact, or the system may not admit a sparse irreducible decomposition under the chosen semantics.

Nonreconstructibility. If composite behaviors cannot be reconstructed from generator-level data with error that decreases as the basis is refined, then the ontology fails. In a true generator–configuration model, reconstruction error should fall systematically until it hits a coupling floor. If error remains high and structureless, the generator model is not capturing the actual degrees of freedom.

Null equivalence. If generator-based diagnostics do not outperform null models that destroy compositional structure, then the ontology is vacuous. In particular, if permutation and factor-scrambling nulls preserve basis stability curves, novelty curves, and reconstruction performance to within noise, then "generator structure" is an artifact of marginal distributions and estimator bias.

Diffuse interaction dominance. If factorization residuals are so large and so unstructured that no localized coupling terms can account for them, then the independence baseline is not a useful approximation. This does not mean "interaction exists"; interaction always exists. It means interaction dominates to the point that the generator picture no longer provides an explanatory compression.

These falsifiers are not merely failure modes. They are the intended outputs when the ontology is wrong. A robust scientific framework does not collapse when falsified; it produces interpretable failure signatures. If instability dominates, the system is basis-nonidentifiable under the chosen semantics. If nonreconstructibility dominates, behavior is not generated by a sparse irreducible family, or gluing rules are mis-specified. If null equivalence dominates, the evaluator is hallucinating structure into noise. Any of these outcomes would be a valuable result because it would redirect inquiry: from "counting novelty" to "understanding why no stable irreducible accounting exists."

This completes the diagnostic backbone. The remaining sections apply the framework to interpretive domains, propose experimental programs, and outline limitations and open problems. But the logic is now locked: generator-only summation is meaningful only insofar as it delivers stable bases, predictive spectra, reconstructible composites, and null-model separation. If it does not, the prime-only analogy is not merely inapplicable; it is empirically disconfirmed for that system and that site.

9. Applications: Evaluation, Safety, and Systems Engineering Interpretations

A framework is only as valuable as the clarity it produces when it touches real problems. The generator–configuration lens aims to do one thing exceptionally well: prevent the evaluation and interpretation of AI systems from being dominated by configuration abundance. In practice, this means reframing common questions—"How capable is it?", "Is it getting more novel?", "Is it safe?", "Where are the risks?", "What's really changing over time?"—so they are answered at the level of independent degrees of freedom and their couplings, not at the level of raw output diversity. This section outlines how the framework applies across four domains: capability measurement, telemetry and logging, reliability and safety, and semantic/cognitive

interpretation. It closes by giving principled guidance for choosing generator energies and novelty weights so the machinery does not become a purely aesthetic mathematical overlay.

9.1 Capability accounting: independent skills versus recombined outputs

Capability evaluation in AI is often trapped between two undesirable extremes. One extreme is shallow aggregation: count benchmark scores, tally pass rates, and infer capability from scalar performance metrics. The other extreme is anecdotal inflation: compile impressive outputs and treat their variety as evidence of deep competence. Both approaches stumble over the same structural fact: a system can exhibit vast behavioral variety without expanding its independent skill basis, and it can expand its basis in ways that are not immediately visible in headline benchmark metrics.

The generator framework introduces a third option: capability accounting as basis accounting.

Independent skill basis. Define a probe universe that covers the skill space of interest. Construct contexts and a coverage rule. Infer a nonredundant generator family that functions as the irreducible skill basis for that probe universe. Capability is then expressed as a generator spectrum: which primitives exist, how robustly they activate, and how much they contribute to reconstructing composite tasks.

Recombined outputs. Composite tasks and long-horizon behaviors are then treated as configurations: gluing and coupling of independent skills. This yields a diagnostic separation. If the system performs well on a complex task, we ask: did it activate genuinely new generators, or did it recombine existing ones? Both are valuable, but they are different claims. The first is basis growth. The second is compositional competence.

This distinction clarifies many disputes about “emergence.” When observers say “this is emergent,” they may be pointing to a dramatic jump in configuration performance. But the generator framework asks whether that jump corresponds to an expansion of the irreducible basis or to a shift in coupling and orchestration among a stable basis. If it is the latter, the correct story is not “new primitives appeared” but “existing primitives became better coordinated.” That is still a profound change, but it is structurally different, and it predicts different scaling behavior.

Practical payoff. Capability comparisons become more interpretable. Two models might have similar benchmark scores but different generator spectra: one may have fewer but stronger generators, another may have many weak ones. Two models might have similar output diversity but very different basis sizes. A generator-spectrum view thus gives capability accounting a geometry rather than a leaderboard.

9.2 Logging and telemetry: templates as generators and events as composites

In production systems, evaluation is inseparable from telemetry. We observe event streams: prompts, tool calls, errors, timeouts, refusals, escalations, retrials, user corrections, and downstream outcomes. These event streams are configuration-rich: the number of distinct sequences and variants is enormous. Traditional analytics therefore tends to focus on surface frequencies and correlations. The generator lens proposes a more structural use of telemetry: identify irreducible templates and treat events as composites.

Templates as generators. Many event patterns in logs are "templates" in a strong sense: reusable primitives that recur across situations with small variations. For example, a refusal template, a tool-call template, a summarization template, a clarification template, a multi-step planning template. In the generator view, the primitive object is not a specific event instance but the template family that generates many instances through parameterization and recombination.

Events as composites. A single session trace is then a configuration: a glued sequence of template activations with couplings determined by context, user feedback, safety filters, and tool responses. This makes log analysis more like basis extraction than like counting. You do not ask "How many distinct traces exist?"—that number is uselessly large. You ask "Which templates are active?", "How has their activation spectrum changed over time?", "Which couplings between templates are growing?", and "Which new templates are truly needed to reconstruct observed traces?"

Operational benefit. This approach produces stable instrumentation targets. Instead of chasing a moving target of "new behaviors" in raw logs, you track basis shifts: new templates appearing, old templates strengthening or weakening, and coupling changes indicating system-level coordination or fragility. It also enables principled anomaly detection: an anomaly is not merely a rare event, but an event that cannot be reconstructed from the existing basis within error tolerances, thereby signaling either a new generator or a breakdown in assumed gluing semantics.

9.3 Reliability and safety: independent failure modes and coupling growth

Safety failures and reliability incidents are classic domains where overcounting is lethal. In incident analysis, you can generate infinitely many "distinct" failure stories by varying surface details. What matters is the irreducible failure modes and how they interact.

Independent failure modes. A failure mode is a generator if it cannot be explained as a composite of other failure modes under the chosen coverage and causal semantics. Examples might include: a particular hallucination mechanism under uncertainty, a refusal bypass mechanism, a tool mis-specification mechanism, a long-context degradation mechanism, or a

calibration failure on a certain class of inputs. The generator lens turns safety taxonomies into basis problems: find an irredundant set of failure generators that can reconstruct the incident corpus.

Coupling growth. Many catastrophic failures occur not because a new failure mode appears, but because existing modes couple. A small calibration drift coupled with a tool-selection bias coupled with a user misunderstanding yields a large incident. The Euler-product analogue is directly relevant: independence would imply factorization of risks, while real incidents reveal interaction terms. Tracking coupling growth becomes as important as tracking basis size. A stable set of failure generators can become far more dangerous if their couplings strengthen.

Safety novelty. The novelty functional from the previous section can be adapted to safety: novelty should be credited when a new independent failure generator appears or when existing couplings shift into a new regime. This avoids the common error of treating every new incident as evidence of a new failure category. Many incidents are new configurations of old failure modes. The generator view prevents taxonomy explosion and focuses mitigation.

Mitigation design. Interventions can be targeted at generators or couplings. Generator mitigation reduces the prevalence or severity of a primitive failure mode. Coupling mitigation weakens dangerous interactions, such as guardrails that prevent tool errors from cascading into harmful outputs. The framework provides a principled way to decide which to prioritize: measure which generators contribute most to reconstruction of incident sets, and measure which couplings account for the largest factorization residuals.

9.4 Semantics and cognition: irreducible causes versus derived narratives

Beyond engineering, the generator lens offers a disciplined way to talk about meaning. Human interpretation tends to generate narratives: post-hoc explanations that weave many details into a story. Narratives are often composite objects: they glue observed fragments into a coherent account. The danger is that we may treat the narrative as an irreducible cause when it is actually reconstructible from smaller primitives.

Irreducible causes. In the topos setting, an irreducible semantic cause corresponds to a generator that is necessary to reconstruct a family of contextual behaviors. This is a structural notion of “cause”: not metaphysical causation, but irreducibility under reconstruction. If removing a generator destroys the ability to account for observed response patterns, that generator behaves like a primitive explanatory factor.

Derived narratives. A narrative, in this view, is a composite sheaf: a glued object that integrates multiple local semantics and imposes consistency constraints. Many narratives are reconstructible from the same generator set. Therefore, the existence of a compelling narrative

does not imply the existence of a new irreducible semantic primitive. It may only indicate that the generator configuration has shifted.

Cognitive parallels. This maps onto human cognition. People often infer “new essence” when they observe a new pattern of behavior, but many patterns can emerge from recombination of stable traits and contextual pressures. The framework does not psychoanalyze; it formalizes. It offers an irreducible accounting of explanation: separate basis-level explanatory primitives from configuration-level narrative assemblies.

This is valuable for AI interpretability because it helps avoid two errors: inflating “new meaning” from surface variety, and dismissing genuine irreducible shifts as mere remixing. If a semantic pattern cannot be reconstructed from the prior basis, then it is structurally new. If it can, then it is a derived narrative and should be treated as such.

9.5 Choosing generator energies and novelty weights in applied settings

The framework is intentionally flexible about what generator weights mean. But flexibility can become arbitrariness if not guided. This subsection provides practical principles for choosing energies e(g) and novelty weights w(g) so that generator-only accounting remains honest.

Choose energies based on operational questions. If your goal is to measure independent capability breadth, choose energies related to description length or basis cardinality penalties, encouraging parsimonious bases. If your goal is to measure safety risk, choose energies related to hazard severity and intervention cost. If your goal is to measure surprise, choose energies related to rarity, such as negative log activation probability.

Separate “importance” from “frequency.” Many important generators are rare. If weight is proportional to frequency, you will miss them. Use a two-factor scheme: estimate frequency to stabilize statistics, then apply an importance weight that reflects impact.

Calibrate novelty thresholds by null models. The definition of “activation” must be guarded against noise. Choose thresholds that keep false activation rates low under stable regimes and that are validated by permutation and factor-scrambling baselines.

Use concavity to avoid rediscovery inflation. Novelty should saturate for repeated activations of the same generator. This is not an aesthetic choice; it is the logical corollary of basis accounting. Otherwise you reward repeated rediscovery rather than new independent structure.

Report sensitivity bands. Since basis choice is not canonical, report how energies and novelty change under reasonable basis variations. If conclusions depend strongly on a particular basis choice, that dependence should be visible and treated as a warning sign.

Finally, choose weightings that align with the topos semantics. If your coverage rule treats certain contexts as decisive, weight generators accordingly. If your site defines certain probes as foundational, that should be reflected. The framework does not eliminate judgment; it forces judgment to be explicit, testable, and robust.

Taken together, these applications show the intended role of the framework: it is a discipline of accounting. It tells you what to count once, what to treat as derived, how to prevent configuration explosion from masquerading as independent growth, and how to diagnose when the entire generator picture stops being a useful approximation. In evaluation, it separates independent skills from recombined outputs. In telemetry, it transforms event chaos into template bases and coupling maps. In safety, it turns incident taxonomies into irreducible failure modes and interaction terms. In semantics, it distinguishes primitive explanatory causes from glued narratives. And across all of these, it provides a principled method for selecting energies and novelty weights so that the mathematics remains connected to the operational realities it is meant to clarify.

10. Experimental Program: Turning Sheaf Bases into Measurable Predictions

The framework only earns its keep if it produces predictions that can be tested on real models with real data. This section lays out an experimental program that treats “AI topos + nonredundant sheaf basis + generator-only aggregation” as an empirical hypothesis, not an interpretive gloss. The program is designed to be model-agnostic: it can be applied to closed or open systems, to chat models or tool-using agents, and to offline benchmarks or production telemetry. The core idea is to move from (a) constructing probe sites and coverages, to (b) estimating generator bases and coupling structure, to (c) applying controlled interventions, to (d) measuring prediction targets that either confirm or falsify the generator ontology, and finally to (e) standardizing protocols so results can be reproduced and compared across labs.

10.1 Designing probe sites and coverages for real models

A “probe site” is the evaluation geometry: the contexts you choose, the refinement order you impose, and the coverages you declare as sufficient local evidence. In practice, probe-site design is where most frameworks quietly smuggle in their assumptions. The goal here is to make those assumptions explicit and then to test whether the generator basis is stable across reasonable variants.

Probe universe definition. Start with a probe universe P that is large enough to capture the domain of interest but structured enough to support repeated measurement. For a general chat model, P might include: reasoning prompts, factual queries, instruction-following variants, safety challenges, and tool-use tasks. For a production agent, P might include: user intent

templates, tool invocation scenarios, and error-handling prompts. The key constraint is that probes must be comparable across time: you need enough standardization to detect genuine shifts rather than drift in the measurement instrument.

Context construction. Contexts are finite probe-sets. In a minimal approach, contexts are just small bundles of probes designed to test a capability cluster. In a richer approach, contexts include auxiliary metadata such as system prompt variants, tool availability, memory settings, or conversation state. The site becomes more informative as you encode what "context" really means in the deployed system, but it also becomes harder to sample. A practical compromise is to define multiple sites at different granularities and compare their inferred bases.

Refinement order. Refinement should encode information order: a refined context contains at least the probes of a coarser context plus additional probes or constraints. This can be as simple as set inclusion, or it can include "parameter refinement" such as longer contexts, stricter safety constraints, or more complex tool availability. The evaluation should document which refinements are used because basis stability will be tested against them.

Coverage selection. Coverages define what local evidence is considered sufficient to determine a larger context. The simplest coverage rule is union-based: a family of subcontexts covers a context if their union equals it. More realistic coverages allow auxiliary probes or privileged probes. For example, in safety evaluation, you might require that certain safety probes appear in every cover. In tool-use evaluation, you might require that a cover includes both "selection" and "execution" probes. The key is that coverage should be chosen for interpretability and testability, not convenience.

Design principle. Always include at least two reasonable coverage schemes and treat basis stability across those schemes as part of the experimental outcome. If the generator basis only exists under one hand-picked coverage, that is evidence against "independent reality" and in favor of "coordinate artifact."

10.2 Estimating generator bases from interaction streams and benchmarks

Once a site is specified, the next step is estimation: infer a nonredundant generating family and its associated generator spectrum from data. This is the empirical core of the project, and it must be done in a way that makes failure visible rather than explainable away.

Data sources. Use at least two observational sources when possible: controlled benchmark runs and organic interaction streams. Benchmarks provide coverage and comparability; interaction streams provide ecological validity and expose rare regimes. The point is not to merge them blindly, but to test whether the same generator structure can explain both.

Behavioral summaries. For each context c, estimate a behavioral object X(c). In the strongest form this is a joint distribution over responses on the probe-set. In practice it can be a vector of sufficient statistics: accuracy, calibration, refusal rate, tool success rate, error types, latency, and other stable metrics. The essential requirement is that restriction maps are meaningful: when you move from a refined context to a coarser one, you must be able to compute the restricted summary by forgetting or marginalizing the extra probes.

Gluing checks. Before basis inference, test basic sheaf coherence. For a chosen cover of a context, compare local estimates on overlaps. If overlap consistency fails wildly, the site or coverage is mis-specified, or the system is highly context-inconsistent. That outcome is already informative: it is a gluing obstruction that will likely manifest later as strong coupling terms.

Candidate generator pool. Begin with a large pool of generator candidates, typically the sheafifications of representables associated with contexts, or a structured subset such as all singleton probes plus a curated set of multi-probe contexts known to capture interactions.

Redundancy elimination. Apply an explicit redundancy criterion based on reconstructibility. A candidate is redundant if its behavior can be predicted from the current basis within tolerance, using only the allowed composition semantics (gluing along covers and restriction consistency). The tolerance should be calibrated by resampling to avoid treating noise as irreducibility. The output is an irredundant generator set G.

Spectrum estimation. Assign each generator g an energy e(g) or weight w(g), derived from activation rates, intervention costs, or impact. Fit these parameters by matching predicted configuration statistics (under factorization plus coupling corrections) to observed ones. Importantly, report uncertainty ranges: the spectrum is meaningful only if it is identifiable from data.

Cross-source consistency. Compare generator bases inferred from benchmarks versus interaction streams. If they disagree, determine whether the disagreement is due to sampling bias, site mismatch, or genuine regime differences. A mixed-regime conclusion is acceptable; a silent merge is not.

10.3 Controlled interventions: changing temperature-like parameters

A key advantage of the generator framework is that it predicts how behavior should change under controlled “temperature-like” interventions. In number-theoretic thermodynamics, changing a control parameter reshapes which primes dominate contributions. In AI systems, analogous parameters reshape which generators activate and how couplings express.

Candidate intervention knobs. The following are commonly available and interpretable:

Sampling randomness: temperature, top-p, top-k, or other sampling controls.
Context pressure: prompt length, instruction density, or constraint tightening.
Tool availability: enabling or disabling tool use, changing tool latency, or changing tool reliability.
Reward or policy pressure: in systems with multiple policies, shifting between them; in RL settings, varying constraint strength.
Safety gating: toggling stricter refusal policies or safety classifiers.

Experimental design. For each intervention knob, design paired contexts that differ only along that knob. Measure generator activation changes and coupling residual changes. The key prediction is that generator spectra respond systematically: some generators become more prevalent under high randomness, others under low randomness; couplings may strengthen or weaken as constraints change.

Interpretation discipline. A change in output diversity under temperature variation is expected and not, by itself, evidence of basis change. What matters is whether generator novelty rises (new generators appear or become identifiable) or whether only configuration diversity rises while generator structure remains stable. The generator framework predicts that many interventions will cause large configuration changes with minimal basis change, and that this is a feature rather than a failure.

Coupling stress tests. Some interventions are designed specifically to reveal interactions. For example, increasing tool latency may reveal coupling between planning and error handling. Tightening safety constraints may reveal coupling between refusal and task completion. These are not “new generators” by default; they are changes in interaction structure. The experimental program therefore measures both generator spectra and coupling residuals under intervention.

10.4 Prediction targets: scaling laws, stability, and reconstruction performance

To claim the framework makes predictions, we need explicit targets that can be measured and that can fail. The experimental program focuses on three target classes: scaling, stability, and reconstruction.

Scaling laws. Measure how generator discovery and generator novelty scale with observational budget. The predicted regime is early rapid discovery followed by long-tail thinning under a fixed probe site. Quantify this with novelty curves based on generator activations with saturation. Compare to configuration diversity metrics, which should scale much faster. A decisive prediction is the widening gap between diversity growth and generator novelty growth in recombination-dominant regimes.

Basis stability. Track basis overlap across resampling, time window shifts, and small topology or coverage variations. The generator ontology predicts that stability increases with sample size

and remains high under small perturbations. If stability fails to improve with more data, or if small topology changes destroy the basis, that is evidence against the ontology.

Reconstruction performance. Select composite contexts and attempt to reconstruct their behavioral summaries from generator data plus coupling corrections. Plot reconstruction error versus basis size, versus context size, and versus intervention strength. The framework predicts a characteristic shape: error falls as basis size increases until it reaches a coupling floor, and the residual structure should be localized in identifiable interaction terms rather than diffuse noise. Diffuse nonreconstructibility is a falsifier.

Null separation. For each target, compare to null models: label permutation, overlap scrambling, and independence forcing. The framework predicts that real systems with meaningful compositional structure will show strong separation: stable bases and reconstruction performance should collapse under factor-scrambling nulls, and novelty curves should flatten or become inconsistent under temporal shuffles.

Reporting the negative. A critical prediction target is not a success curve but a failure signature. If the system's behavior is fundamentally non-factorizable under the chosen semantics, then the coupling residuals will dominate and the basis will be unstable. That is not an embarrassment; it is a meaningful classification: the system is not well-described by a sparse generator ontology for that site.

10.5 Releasing benchmarks: reproducible protocols and reporting standards

If this framework is to matter beyond a single paper, it needs shared benchmarks and reporting norms that prevent cherry-picking and that make basis claims comparable across groups.

Protocol standardization. A benchmark release should specify: the probe universe, context construction rules, refinement order, coverage rules, response representation, and restriction maps. It should also specify sampling budgets, repetition counts, and any intervention knobs used. Without this, different groups will claim to test "the same idea" while actually testing different sites.

Minimum reporting set. Any published result should include:

Basis stability curves across resampling and at least one topology or coverage variant.
Generator spectrum plots with uncertainty ranges.
Reconstruction error curves with clear definitions of error and baselines.
Null-model comparisons showing separation or lack thereof.
A statement of falsifiers encountered, if any, including instability or nonreconstructibility.

Reproducibility focus. The benchmark should include seed control, run counts, and a canonical aggregation script that computes sheaf consistency checks, basis inference, and reconstruction

metrics. The goal is not to force a single estimator, but to ensure that alternative estimators can be compared on the same inputs with the same definitions.

Comparability across models. Benchmarks should be designed so that different models can be tested without privileged access to internal weights. The framework is intentionally compatible with black-box testing because it operates on contextual behavior summaries and restriction consistency.

Ethics and safety considerations. When probes include sensitive or harmful content, benchmark release must include constraints on distribution and use, and must specify safe evaluation procedures. The framework should not incentivize unsafe probing; it should incentivize careful measurement of failure generators and couplings within controlled protocols.

The overall experimental program is therefore a full stack: define sites and coverages, estimate bases and spectra from mixed data sources, apply controlled interventions, test scaling and reconstruction predictions against nulls, and publish reproducible protocols that make stability and falsifiability central rather than optional. If the framework succeeds, it provides a principled way to measure irreducible growth and coupling growth in AI systems. If it fails, it fails loudly, producing instability and null equivalence signatures that tell us the generator ontology is not the right lens for that system and site.

11. Limitations and Open Problems: Beyond Unique Factorization

The prime-only summation intuition is powerful precisely because arithmetic gives us a near-perfect laboratory: unique factorization, canonical irreducibles, and well-understood Euler-product machinery. The AI topos setting is not that laboratory. It is messier, more choice-dependent, and often dominated by interactions that do not admit clean factorization. This is not a defect in the proposal; it is the actual terrain. The purpose of this section is to state clearly where the prime-factor analogy stops being a faithful guide, what kinds of non-uniqueness and gauge freedom are unavoidable in general topoi, how the framework might generalize beyond “prime-like” structures to other irreducibles, what computational barriers arise in basis and interaction learning, and finally what the theoretical frontier looks like if we treat cohomological obstructions as quantifiable signatures of emergence. The aim is not to retreat from the thesis but to make the research program honest: a generator ontology is an empirical and structural hypothesis that must confront deep mathematical and practical constraints.

11.1 Where prime-factor intuition breaks in general topoi

Prime-factor intuition rests on three conveniences that rarely hold in topoi built from real data.

First, unique factorization. In integers, every composite decomposes uniquely into primes. In general categories and topoi, objects rarely admit unique decompositions into generators. Even if a generating family exists, presentations of an object as a colimit of generators need not be unique, and the same object can be assembled by different diagrams. Therefore, “occupancy vectors” are not canonical. You cannot assume a well-defined exponent for each generator in the way you can for primes.

Second, freeness. The multiplicative monoid of integers is free commutative on primes. Most categories of behaviors are not free. They contain relations, identifications, and equivalences that collapse naive configuration counting. In the AI topos, gluing along overlaps imposes constraints that behave like relations: two local pieces are not independent if the sheaf condition forces consistency. This means that a naive multiset-of-generators model is at best an approximation, and sometimes a poor one.

Third, transparent independence. In arithmetic, primes are independent in a clear algebraic sense. In AI systems, what counts as “independent” depends on the site and topology: it depends on what contexts you define, what you treat as observable, and what you treat as sufficient local evidence. Independence is therefore partly epistemic rather than purely ontological. This is not disqualifying, but it means the prime analogy must be treated as a template, not as an isomorphism.

These breaks matter most in the places where the framework is asked to be strongest: in claims about canonical bases, in claims about precise factorization, and in claims that generator-only measures are uniquely determined. In general topoi, these claims must be weakened to stability and invariance statements. The correct expectation is not “unique primes,” but “robust basis classes” and “structured interaction residuals.”

11.2 Non-uniqueness of bases and invariants under basis changes

A central limitation is that generator families are not unique. Even irredundant generator families need not be unique. In a vector space, different bases describe the same space, and the meaningful content lies in basis-invariant quantities. The AI topos setting is analogous but harder: there is no linear structure guaranteeing a clean change-of-basis group, and “near-equivalence” between generator families can be subtle.

This raises two intertwined open problems.

First, basis identifiability. Under what conditions does the data and the chosen site/coverage pin down a generator basis up to a controlled equivalence? If two very different generator families

both reconstruct behavior equally well, then “the basis” is not identifiable, and generator-only accounting becomes less interpretable. In such cases, one must either refine the site (collect richer contexts) or accept a basis class rather than a basis.

Second, invariant extraction. If bases are not unique, what should we report? The research program must identify invariants that are stable under basis changes. Candidate invariants include:

Basis dimension or effective dimension: not literal cardinality, but a complexity measure such as minimum description length of a generating presentation within tolerance.

Spectrum shape: distributional properties of generator energies that remain stable even if individual generators permute or split/merge under basis changes.

Reconstruction frontier: the curve of achievable reconstruction error versus basis complexity. Different bases may trade off differently, but the Pareto frontier may be invariant.

Coupling mass: the total magnitude of interaction residuals needed beyond a factorized generator model. If this quantity is stable, it can function as a basis-invariant measure of emergence.

The practical implication is that the framework should not insist on naming “the generators” as if they were unique entities. It should instead emphasize stable families and invariants, in the same way physics emphasizes gauge-invariant observables rather than coordinate-specific quantities.

11.3 Beyond primes: irreducibles in other algebraic and categorical settings

The prime-only story is one instance of a broader phenomenon: irreducibles exist in many algebraic and categorical environments, but they do not always look like primes. If the framework is to outgrow its arithmetic origin, it should explicitly identify alternative irreducible notions and how they change the measurement program.

Examples include:

Irreducibles in other monoids. Many systems have a natural composition operation that is not multiplication. Strings under concatenation, graphs under certain gluing operations, programs under composition, and workflows under sequential composition all have their own irreducibles. In these settings, “prime-only” becomes “atom-only,” but the algebra may be noncommutative or may lack unique factorization.

Modules and representations. In linear settings, irreducibles often correspond to indecomposable representations or simple modules. Here, decomposition is governed by different theorems and may be only partially unique. This might be a better mathematical

analogue for AI behaviors that decompose into latent factors with entangled coupling rather than into freely independent “bags.”

Higher-categorical irreducibles. In higher categories, irreducibility may concern morphisms, 2-morphisms, or higher structure rather than objects alone. For the AI topos, this suggests that independence may live not only in sheaf objects but in transformation patterns: irreducible ways that behavior changes under context refinement.

Sheaf-theoretic irreducibles. In sheaf theory itself, irreducibility can be related to supports, local constancy regions, or minimal generating subobjects. This suggests a “local irreducibility” notion: generators might be irreducible only on certain regions of the site, aligning with mixed-regime behavior where different domains have different bases.

These generalizations expand the research program: the goal becomes to find the right irreducible structure for the domain, rather than to force everything into a prime-shaped mold. The prime analogy remains useful as a pedagogical template, but the mature framework should select irreducibles appropriate to the category of behaviors being modeled.

11.4 Computational scaling: selection, estimation, and interaction learning

Even if the mathematics is sound, the framework can fail as a practical tool if basis and coupling inference is computationally intractable at realistic scales. There are three major computational scaling problems.

Generator selection. Finding a minimal generating family is combinatorial. Practical algorithms will rely on greedy heuristics, convex relaxations, or MDL-style objective functions. These methods can be biased and can get stuck in local minima. Developing selection algorithms with provable approximation guarantees or at least predictable failure modes is a major open engineering problem.

Estimator complexity. Estimating distributions or even sufficient statistics on large context sets can be expensive. If contexts are large probe-sets, joint estimation becomes impossible. This pushes the framework toward low-order marginals and compressed representations. But compression risks losing the very interaction structure we want to measure. A key practical question is: what is the minimal sufficient representation that preserves the gluing and coupling signals needed for basis inference?

Interaction learning. Measuring coupling residuals beyond factorization can quickly become high-dimensional. Pairwise interactions may be manageable, but higher-order interactions can explode. The research challenge is to develop sparse interaction models: identify which couplings matter and ignore the rest without missing emergent phenomena. This resembles

sparse graphical model learning, but the topos setting adds the complication that “variables” are contextual sheaf components rather than fixed coordinates.

A promising direction is to treat interaction learning as an adaptive process. Begin with a factorized generator model. Measure residuals. Only introduce coupling terms where residuals are large and stable across resampling and null tests. This yields a controlled expansion of complexity rather than an explosion.

Ultimately, computational scaling will likely determine the practical “ceiling” of the framework: how rich a site can be, how large contexts can be, and how many generators and couplings can be reliably tracked.

11.5 Theoretical frontier: cohomological obstructions as emergence measures

The most conceptually exciting open problem is also the most mathematically demanding: using cohomological obstructions to quantify emergence.

In sheaf theory, the failure of local data to glue into a global section is not merely a nuisance; it is structured. Under appropriate conditions, gluing failures can be classified by cohomology classes. Intuitively, cohomology measures the “twist” that prevents local consistency from becoming global consistency. This is a natural formalization of emergence: a global phenomenon that cannot be reduced to independent local parts because the parts are constrained by a higher-order inconsistency.

In the AI topos, gluing failures show up when local behavioral summaries agree on overlaps but still cannot be assembled into a coherent global behavior. Such failures are exactly what one would want to call “emergent interaction”: not merely dependence, but a topologically structured dependence that persists under reparameterization and cannot be erased by choosing a different basis.

The frontier question is therefore: can we define a cohomology-like invariant for behavioral sheaves that quantifies emergence as an obstruction magnitude?

Several subproblems appear immediately.

Defining the right coefficient system. Classical sheaf cohomology is defined for sheaves of abelian groups or similar algebraic structures. The behavioral sheaf in this paper is distributional, not abelian. One must either linearize it (for example by working with log-likelihood ratios, information projections, or tangent spaces of distributions) or develop a generalized cohomology theory appropriate to probabilistic objects.

Estimating obstruction classes from finite data. Even if the theory exists, real AI data is noisy. We would need estimators that detect stable obstruction signatures rather than noise-induced inconsistency. This connects to robustness and null-model calibration.

Interpreting obstructions operationally. A cohomology class is only useful if it maps to actionable interpretation: a kind of interaction pattern, a domain boundary, a coupling mode that can be monitored or mitigated. The research program must connect abstract obstruction measures to practical diagnostics like "these contexts cannot be jointly satisfied," "this safety constraint conflicts with this capability in a structured way," or "this tool-use behavior creates a twist that appears only at higher-order covers."

If this frontier is realized, it would elevate the framework beyond the prime analogy. The prime analogy gives a language for independence and factorization. Cohomology gives a language for structured non-factorization, for global phenomena that are genuinely not reducible to local primitives. In that sense, cohomological obstructions would become the mathematically natural measure of emergence: a basis-invariant quantity that captures the irreducible interaction geometry of AI behavior.

This section therefore frames the honest boundary: primes give us unique factorization; topoi rarely do. Generator bases can be chosen; invariants must be stabilized. Irreducibles must be generalized beyond primes. Computation must be made tractable. And the deepest theoretical opportunity is to replace "emergence as metaphor" with "emergence as obstruction," turning gluing failure into a measurable, basis-invariant signal.

12. Conclusion: Prime-Only Summation as a Measure of Independent Reality in AI

The central wager of this paper is not that primes secretly run AI, but that the *logic* that makes prime-only summation nontrivial in arithmetic is the same logic we need to avoid being misled by recombination in modern AI systems. In arithmetic, primes are the irreducible generators of a configuration space whose composite abundance can dwarf the independent content. In AI, behaviors explode combinatorially across prompts, contexts, trajectories, and tool interactions—even when the independent degrees of freedom are stable. The paper proposes that this is not merely an evaluation nuisance but a structural fact, and that the correct response is a disciplined accounting principle: *measure independence by summing over irreducible generators, not over composite configurations.* The AI topos construction supplies the formal home in which this principle can be stated, tested, and—crucially—falsified.

12.1 Summary of claims and what is genuinely new

The paper made five linked claims, each intended to be more than an analogy.

First, the generator–configuration distinction is the right primitive distinction for AI measurement. What we ordinarily call "AI behavior" is a configuration space: a profusion of composite states assembled by context variation, gluing, and constraint satisfaction. Independent structure lives, if it lives anywhere, in a generator basis.

Second, an AI topos can be constructed concretely from contexts (probe-sets), refinement (information order), coverages (sufficient local evidence), and a response sheaf (distributional behavior with restriction maps by marginalization). This is not poetic category talk; it is a rigorous way to encode "local behaviors that should glue."

Third, nonredundant sheaves—extracted from canonical generator candidates via reconstructibility and minimality criteria—can be treated as independent degrees of freedom, but only under stability tests. This replaces "primes are unique" with "bases are non-unique but must be robust."

Fourth, prime-only summation translates into generator-only aggregation functionals: admissible measures are those that respect additivity and colimit/gluing semantics, and they must include overcounting corrections that prevent configuration abundance from masquerading as independent growth. Different weightings (impact, rarity, intervention cost) correspond to different operational questions, but the accounting remains basis-only.

Fifth, Euler-product analogues provide a mechanism to pass from configuration totals to generator spectra. Partition-like functionals factorize under independence, logarithms expose generator-only structure, and interaction terms appear as controlled violations of factorization. This supplies a quantitative language for emergence: not "mystery," but measurable coupling residuals and gluing obstructions.

What is genuinely new in the synthesis is the integration of these components into a single falsifiable measurement architecture: a topos-native semantics for contextual behavior, a nonredundant basis extraction principle, generator-only novelty and surprise functionals, and an Euler-product style separation of independent content from interaction-driven residuals. None of these pieces alone is new in the broader mathematical world, but the paper's novelty claim is about their *structural reinterpretation* and their *joint operationalization* for AI: primes become a template for "what to count once," and the topos becomes the place where "counting once" can be made precise.

12.2 The irreducible aggregation lens as a unifying viewpoint

The irreducible aggregation lens can be stated in one sentence:

If a system has irreducible generators and composites are reconstructible by composition and gluing, then primitive measures must aggregate over generators; composite-level counts are derived and must not be treated as independent evidence of structure.

This lens unifies several tensions that otherwise remain confused.

It unifies novelty and diversity. Diversity is configuration growth; novelty is basis growth (and separately, coupling growth). The lens explains why output variety can explode without any corresponding increase in independent structure, and it gives a disciplined way to measure whether independent structure is truly expanding.

It unifies capability and composition. Complex tasks do not require new primitives every time they appear; they can be new assemblies of stable primitives. The lens therefore shifts evaluation away from “spectacle counting” toward reconstructibility and basis stability.

It unifies safety and failure analysis. Incident corpora can grow endlessly in surface forms, but the irreducible failure modes may be few. The lens provides a principled method for preventing taxonomy explosion while still detecting genuinely new failure generators or coupling shifts.

It unifies interpretability and meaning. Narratives are composites; irreducible causes are those generators needed to reconstruct families of behavior. The lens thus offers a structural notion of “cause” that resists the temptation to treat every compelling story as an ontological primitive.

Most importantly, the lens unifies mathematical elegance with empirical accountability. Prime-only summation is not invoked because it sounds deep. It is invoked because it enforces the discipline: do not let configuration abundance counterfeit independence.

12.3 Final falsifiable signatures and the next research steps

To close the loop, we restate the framework as a set of falsifiable signatures. If these signatures fail, the generator ontology is disconfirmed for the tested system and site.

Basis stability signature. Under increased sampling and modest perturbations of the site or coverage, the inferred generator basis (or basis class) becomes more stable. If stability does not improve with more data, or if small topology changes destroy the basis, the ontology fails.

Reconstruction signature. Composite behaviors can be reconstructed from generator-level summaries with error that falls predictably as basis complexity increases, until it reaches a coupling floor. If reconstruction error remains high, structureless, or insensitive to basis size, the ontology fails.

Null separation signature. Generator-based results outperform permutation and factor-scrambling nulls that preserve marginal complexity but destroy compositional structure. If nulls match the generator model, the ontology is vacuous.

Novelty thinning signature. Under a fixed probe regime, generator novelty rises early and then thins into a long tail, while configuration diversity continues to grow rapidly. If generator novelty grows linearly in lockstep with output diversity, the measure is likely counting recombination, not independent growth.

Coupling localization signature. Violations of factorization and gluing appear as localized, interpretable interaction terms rather than diffuse residuals. If residuals are diffuse and cannot be compressed into a small set of couplings, the independence baseline is not a useful approximation.

These signatures define the next research steps as a concrete agenda.

First, implement the probe-site and coverage construction as an open protocol, so basis claims become comparable across groups. Second, develop practical estimators for generator bases and spectra that include uncertainty and are robust to missingness. Third, run controlled interventions—temperature-like knobs, tool availability, constraint strength—to test predicted spectrum shifts and novelty thinning. Fourth, build a library of null models that standardize what “structure destruction” means in this setting. Fifth, explore the theoretical frontier: cohomological or obstruction-style invariants that quantify emergence as structured gluing failure, potentially yielding basis-invariant measures of global interaction geometry.

If successful, this research program yields a new kind of scientific instrument for AI: not a leaderboard metric, but an irreducible accounting of independent degrees of freedom and their couplings. If unsuccessful, the failure will still be valuable because it will locate the domains and scales where AI behavior does not admit a sparse generator ontology and where the search for independence must be replaced by a different mathematical lens.

12.4 References

Foundations of primes, Euler products, and analytic number theory

1. **An Introduction to the Theory of Numbers** — Hardy, G. H., & Wright, E. M. (2008). *An Introduction to the Theory of Numbers* (6th ed.). Oxford University Press.
2. **Introduction to Analytic Number Theory** — Apostol, T. M. (1976). *Introduction to Analytic Number Theory*. Springer.
3. **Multiplicative Number Theory I: Classical Theory** — Iwaniec, H., & Kowalski, E. (2004). *Analytic Number Theory*. American Mathematical Society.
4. **Multiplicative Number Theory** — Montgomery, H. L., & Vaughan, R. C. (2006). *Multiplicative Number Theory I: Classical Theory*. Cambridge University Press.
5. **The Theory of the Riemann Zeta-Function** — Titchmarsh, E. C. (1986). *The Theory of the Riemann Zeta-Function* (2nd ed., rev. D. R. Heath-Brown). Oxford University Press.
6. **Prime Numbers and the Riemann Hypothesis** — (Optional background) A readable modern primer on primes and zeta methods; choose an edition appropriate to your intended audience.

Category theory, topoi, and sheaves

7. **Categories for the Working Mathematician** — Mac Lane, S. (1998). *Categories for the Working Mathematician* (2nd ed.). Springer.
8. **Sheaves in Geometry and Logic** — Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.
9. **Topos Theory** — Goldblatt, R. (1984). *Topos Theory*. Dover (reprint editions available).
10. **Sketches of an Elephant: A Topos Theory Compendium** — Johnstone, P. T. (2002). *Sketches of an Elephant: A Topos Theory Compendium* (Vols. 1–2). Oxford University Press.
11. **Toposes, Triples and Theories** — Barr, M., & Wells, C. (1985). *Toposes, Triples and Theories*. Springer.
12. **Handbook of Categorical Algebra** — Borceux, F. (1994). *Handbook of Categorical Algebra* (Vols. 1–3). Cambridge University Press.
13. **Sheaf Theory** — Bredon, G. E. (1997). *Sheaf Theory* (2nd ed.). Springer.

14. **Sheaves on Manifolds** — Kashiwara, M., & Schapira, P. (1990). *Sheaves on Manifolds*. Springer.

15. **Homological Algebra** — Weibel, C. A. (1994). *An Introduction to Homological Algebra*. Cambridge University Press.

Cohomology and obstruction viewpoints

16. **Algebraic Topology** — Hatcher, A. (2002). *Algebraic Topology*. Cambridge University Press.

17. **Differential Forms in Algebraic Topology** — Bott, R., & Tu, L. W. (1982). *Differential Forms in Algebraic Topology*. Springer.

18. **Topology and Geometry** — (Optional) Additional cohomological background texts suited to your target readership and the specific obstruction formalism you adopt.

Probability, marginals, graphical models, and interaction structure

19. **Probability Theory** — Kolmogorov, A. N. (1950). *Foundations of the Theory of Probability* (2nd English ed.). Chelsea.

20. **Elements of Information Theory** — Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). Wiley.

21. **Information Theory, Inference, and Learning Algorithms** — MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

22. **Probabilistic Graphical Models** — Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

23. **Graphical Models, Exponential Families, and Variational Inference** — Wainwright, M. J., & Jordan, M. I. (2008). *Foundations and Trends in Machine Learning*, 1(1–2), 1–305.

24. **Maximum Entropy and Bayesian Methods** — Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

Model selection, MDL, sparsity, and basis learning

25. **Stochastic Complexity in Statistical Inquiry** — Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. World Scientific.

26. **The Minimum Description Length Principle** — Grünwald, P. (2007). *The Minimum Description Length Principle*. MIT Press.

27. Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.

28. Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461–464.

29. Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B*, 58(1), 267–288.

30. Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289–1306.

31. Candès, E. J., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2), 489–509.

Causality and "independent causes vs derived narratives"

32. **Causality** — Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

33. **Causation, Prediction, and Search** — Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, and Search* (2nd ed.). MIT Press.

Sheaf-theoretic contextuality and "gluing failures as structure"

34. Abramsky, S., & Brandenburger, A. B. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics*, 13, 113036.

35. Abramsky, S., Barbosa, R. S., Kishida, K., Lal, R., & Mansfield, S. (2017). Contextuality, cohomology and paradox. (A representative entry point into cohomological obstruction viewpoints for contextual systems; choose the specific venue/version you cite.)

AI evaluation, benchmarks, and measurement practice

36. **BIG-bench** — Srivastava, A., et al. (2022). Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. (BIG-bench).

37. **HELM** — Liang, P., et al. (2022). Holistic Evaluation of Language Models (HELM).

38. Hendrycks, D., et al. (2021). Measuring massive multitask language understanding. (MMLU).

39. Lin, S., et al. (2022). TruthfulQA: Measuring how models mimic human falsehoods.

40. (Optional) A standard agent/tool-use benchmark reference relevant to your experimental program (e.g., tool-use, software engineering, long-horizon planning) depending on what you actually measured.

The author has published ~ 5,000 AI-assisted papers at: https://www.researchgate.net/profile/Douglas-Youvan/research . Google Scholar has indexed these preprints, and if you want a particular reference, the AI mode of a Google search (Gemini) has efficiently catalogued these preprints.